HIGHER NEWTON POLYGONS IN THE COMPUTATION OF DISCRIMINANTS AND PRIME IDEAL DECOMPOSITION IN NUMBER FIELDS

JORDI GUÀRDIA, JESÚS MONTES, AND ENRIC NART

Abstract. We present an algorithm for computing discriminants and prime ideal decomposition in number fields. The algorithm is a refinement of a p-adic factorization method based on Newton polygons of higher order. The running-time and memory requirements of the algorithm appear to be very good: for a given prime number p, it computes the p-valuation of the discriminant and the factorization of p in a number field of degree 1000 in a few seconds, in a personal computer.

1. Introduction

The factorization of prime numbers in number fields is a classical problem, whose resolution lays at the foundation of algebraic number theory. Although it is completely understood from the theoretical point of view, the rising of computational number theory in the last decades has renewed the interest on the problem from a practical perspective. In his comprehensive book [Coh00, p. 214], H. Cohen refers to this problem as one of the main computational tasks in algebraic number theory.

The most common insight in the known solutions of the problem is based on the solution on a more general problem: the determination of a (local) integral basis. There is a number of highly efficient methods for this problem, due to H. Zassenhaus and M. Pohst [PZ89], D. Ford and P. Letard [FL94], and D. Ford, S. Pauli and X. Roblot [FPR02].

The theory of higher order Newton polygons developed in [Mon99] and revised in [HN], HN standing for "higher Newton", has revealed itself as a powerful tool for the analysis of the decomposition of a prime p in a number field. Higher Newton polygons are a p-adic tool, and their computation involves no extension of the ground field, but only extensions of the residue field; thus, they constitute an excellent tool for a computational treatment of the problem. In this paper we explain how the theoretical results of [HN] apply to yield an algorithm, due to the second author [Mon99, Ch.3], to factor a prime number p in a number field K, in terms of a generating equation $f(x)$. The algorithm computes the p-valuation of the index of $f(x)$ as well; in particular, it determines the discriminant of the number field, once one is able to factorize the discriminant $\operatorname{disc}(f)$ of the defining equation.

In many applications, the computation of an integral basis is very useful because it helps to carry out other tasks in the number field. However, if one is interested only in the discriminant or in the factorization of a prime, our direct method has the advantage of being more efficient and it makes possible to carry out these tasks in number fields of much higher degree. In fact, the running-time and memory requirements of the algorithm appear to be very good. Even in some bad cases, chosen to test the limit of its capabilities, it computes the factorization of p in a number field of degree 1000 and p-index 200000 in a few seconds, in a personal computer.

2000 *Mathematics Subject Classification.* Primary 11R04; Secondary 11R29, 11Y40.

Key words and phrases. number field, Newton polygon, prime ideal decomposition, discriminant, index.

Partially supported by MTM2006-15038-C02-02 and MTM2006-11391 from the Spanish MEC.

If we add the computation of generators of the prime ideals, the running-time may increase in a significant way, because this routine implies an extended gcd computation.

The outline of the paper is as follows. In section 2 we present the basic algorithm that is obtained by a direct application of the ideas of [HN]. In section 3 we introduce an optimization based on a lowering of the order in which the computations take place, and we prove a strong optimization result (Theorem 3.1). We refer to this optimization process as *refinement*, and it results in a dramatic lowering of the complexity. In section 4 we show how to compute generators of the prime ideals lying above p in terms of the output of the algorithm. In section 5 we describe an implementation, and in section 6 we present the results of some numerical tests. We construct some "worse possible" polynomials, that should be specially difficult with respect to the structure of the algorithm; this means that they have a huge index, and this index is sufficiently "hidden" to force the algorithm to work in a high order. The record is a polynomial of degree 6912 and 2-index 77673504, for which the factorization of 2 is obtained in 787 seconds. The algorithm, moreover, is highly parallelizable, so that it can raise the bounds of computations on number fields to huge degrees.

The local nature of all the computations involved in the algorithm justifies its high efficiency compared to the classical insight explained above. Anyway, after this algorithm, one can go the other way round and apply it as a previous step in the determination of an integral basis. Numerical experimentation suggests that this new approach provides a significant improvement in the solution of this problem.

2. Computation of discriminants and prime ideal decomposition in number fields

We fix a number field $K=\mathbb{Q}(\theta)$, generated by a monic irreducible polynomial $f(x)\in\mathbb{Z}[x]$, such that $f(\theta)=0$. We denote by $\mathbb{Z}_K$ the ring of integers of K. We fix also a prime number $p\in\mathbb{Z}$. The p-adic valuation is denoted simply by v in order to avoid confusion with p-adic valuations v_r of higher order. If $\mathbb{F}$ is a finite field and $\varphi(y)$, $\psi(y)\in\mathbb{F}[y]$, we write $\varphi\sim\psi$ to indicate that the two polynomials coincide up to multiplication by a nonzero constant in $\mathbb{F}$.

In this section we present the basic algorithm that computes the p-value of the discriminant of K and the prime ideal decomposition of $p\,\mathbb{Z}_K$, that is obtained by a direct application of the ideas of [HN].

2.1. Types and their representatives.

The basic tool for the algorithm is the concept of type and its representative, which we recall here with some detail. All results of this section are taken from [HN, §2].

Definition 2.1. *A type of order zero is a monic irreducible polynomial in $\mathbb{F}_p[y]$. Let $r\ge 1$ be a natural number. A* type of order r *is a sequence of data:*

$$\mathbf{t}=(\phi_1(x);\lambda_1,\phi_2(x);\dots;\lambda_{r-1},\phi_r(x);\lambda_r,\psi_r(y)),$$

where $\phi_1(x),\dots,\phi_r(x)\in\mathbb{Z}[x]$ are monic polynomials, $\lambda_1,\dots,\lambda_r\in\mathbb{Q}^-$ are negative rational numbers, and $\psi_r(y)\in\overline{\mathbb{F}}_p[y]$ is a monic polynomial, that satisfy the following properties:

1. *$\phi_1(x)$ is irreducible modulo p. Let $\psi_0(y)\in\mathbb{F}_p[y]$ be the polynomial obtained by reduction of $\phi_1(y)$ modulo p. We define $\mathbb{F}_1:=\mathbb{F}_p[y]/(\psi_0(y))$.*
2. *For all $1\le i<r$, the Newton polygon of i-th order, $N_i(\phi_{i+1})$, is one-sided, with positive length and slope λ_i.*
3. *For all $1\le i<r$, the residual polynomial of i-th order with respect to λ_i, $R_i(\phi_{i+1})(y)$, is an irreducible polynomial in $\mathbb{F}_i[y]$. Let $\psi_i(y)\in\mathbb{F}_i[y]$ be the monic polynomial determined by $R_i(\phi_{i+1})(y)\sim\psi_i(y)$. We define $\mathbb{F}_{i+1}:=\mathbb{F}_i[y]/(\psi_i(y))$.*
4. *$\psi_r(y)\in\mathbb{F}_r[y]$ is a monic irreducible polynomial, $\psi_r(y)\ne y$. We define $\mathbb{F}_{r+1}:=\mathbb{F}_r[y]/(\psi_r(y))$.*

Every type carries implicitly a certain amount of extra data, whose notation we fix now. For all $1 \le i \le r$:

- h_i, e_i are a pair of positive coprime integers such that $\lambda_i = -h_i/e_i$,
- $\ell_i,\, \ell'_i \in \mathbb{Z}$ are fixed integers such that $\ell_i h_i - \ell'_i e_i = 1$,
- $f_i = \deg \psi_i(y)$, and $f_0 = \deg \psi_0(y) = \deg \phi_1(x)$,
- $m_i = \deg \phi_i(x)$, and $m_{r+1} = m_r e_r f_r$. Note that $m_{i+1} = e_i f_i m_i$,
- $z_i = y \pmod{\psi_i(y)} \in \mathbb{F}_{i+1}^*$, $z_0 = y \pmod{\psi_0(y)} \in \mathbb{F}_1^*$. Thus, $\mathbb{F}_{i+1} = \mathbb{F}_i(z_i)$.

Also, for all $1 \le i \le r+1$, the type carries certain p-adic discrete valuations $v_i : \mathbb{Q}_p(x)^* \to \mathbb{Z}$ [HN, Def.2.5], and semigroup homomorphisms,

$$\omega_i \colon \mathbb{Z}_p[x] \setminus \{0\} \to \mathbb{Z}_{\ge 0}, \qquad P(x) \mapsto \operatorname{ord}_{\psi_{i-1}}(R_{i-1}(P)),$$

where $R_0(P)(y) \in \mathbb{F}_p[y]$ is the reduction modulo p of $P(y)/p^{v(P)}$. These objects play an essential role in what follows, because $\omega_i(P)$ measures the length of the principal part, $N_i^-(P)$, of the Newton polygon of i-th order of $P(x)$ [HN, Lem.2.17]. The principal part N^- of a polygon N is the polygon determined by the sides of negative slope of N.

To avoid confusion, in case of working simultaneously with different types, we add a superscript with the type to every component or datum: $\phi_i^{\mathbf{t}}(x)$, $\lambda_i^{\mathbf{t}}$, $e_i^{\mathbf{t}}$, etc.

Definition 2.2. *We say that $\lambda_i, \phi_{i+1}(x)$ (and their implicit data) are the i-th* level *of $\mathbf{t}$.*

By truncation we can easily obtain types of lower order. We denote

$$\mathbf{t}_i := (\phi_1(x); \lambda_1, \phi_2(x); \cdots ; \lambda_{i-1}, \phi_i(x); \lambda_i, \psi_i(y)), \quad 0 \le i \le r,$$

$$\tilde{\mathbf{t}}_i := (\phi_1(x); \lambda_1, \phi_2(x); \cdots ; \lambda_{i-1}, \phi_i(x); \lambda_i, \phi_{i+1}(x)), \quad 0 \le i < r.$$

Clearly, $\mathbf{t}_i$ is a type of order i. The *extension* $\tilde{\mathbf{t}}_i$ is not a type, strictly speaking.

To our polynomial $f(x) \in \mathbb{Z}[x]$ we can attach the set $\mathbf{t}_0(f)$ of all types of order zero that divide $f(x)$ modulo p. By Hensel's lemma, each $\mathbf{t} \in \mathbf{t}_0(f)$ determines a monic p-adic factor $f_{\mathbf{t}}(x) \in \mathbb{Z}_p[x]$ of $f(x)$, and

$$f(x) = \prod_{\mathbf{t} \in \mathbf{t}_0(f)} f_{\mathbf{t}}(x).$$

The types of order r play an analogous role and they provide similar factorizations in higher order. Let us recall some concepts and results in this respect.

Definition 2.3. *Let $\mathbf{t}$, $\mathbf{t}'$ be types of order r, and let $P(x) \in \mathbb{Z}_p[x]$ be a monic polynomial.*

- *We say that $P(x)$ has type $\mathbf{t}$ if $\deg P = m_{r+1}\omega_{r+1}(P) > 0$, or equivalently*
  (1) *$P(x) \equiv \phi_1(x)^{a_0} \pmod p$, for some positive integer a_0, and*
  (2) *For all $1 \le i \le r$, the Newton polygon $N_i(P)$ is one-sided, of slope λ_i, and $R_i(P)(y) \sim \psi_i(y)^{a_i}$ in $\mathbb{F}_i[y]$, for some positive integer a_i.*
- *We say that $P(x)$ is divisible by $\mathbf{t}$, or that $\mathbf{t}$ divides $P(x)$, if $\omega_{r+1}^{\mathbf{t}}(P) > 0$. Formally, we can think of $\omega_{r+1}^{\mathbf{t}}(P)$ as the exponent with which $\mathbf{t}$ divides $P(x)$.*
  If $\mathbf{t}$ divides $P(x)$, we denote by $P_{\mathbf{t}}(x)$ the monic factor of $P(x)$ of type $\mathbf{t}$ and greatest degree. It has $\deg P_{\mathbf{t}} = m_{r+1}\omega_{r+1}^{\mathbf{t}}(P)$, and $\omega_{r+1}^{\mathbf{t}}(P_{\mathbf{t}}) = \omega_{r+1}^{\mathbf{t}}(P)$.
- *We say that $\mathbf{t}$ and $\mathbf{t}'$ are P-equivalent, if both divide $P(x)$, and $P_{\mathbf{t}}(x) = P_{\mathbf{t}'}(x)$.*
- *We say that a set $\mathbf{T}$ of types faithfully represents $P(x)$, if $P(x)$ is divisible by all types in $\mathbf{T}$, and $P(x) = \prod_{\mathbf{t} \in \mathbf{T}} P_{\mathbf{t}}(x)$.*

In [HN, §2.3] it is described a constructive method to enlarge a type of order r into different types of order $r+1$.

Theorem 2.4. *Let $\mathbf{t}$ be a type of order r. We can effectively construct a monic polynomial $\phi_{r+1}(x) \in \mathbb{Z}[x]$ of type $\mathbf{t}$ such that $\omega_{r+1}(\phi_{r+1}) = 1$. This polynomial has minimal degree $\deg \phi_{r+1} = m_{r+1}$ among all polynomials of type $\mathbf{t}$.*

We call such a polynomial $\phi_{r+1}(x)$ a *representative of the type* $\mathbf{t}$. We denote by $\tilde{\mathbf{t}} := (\phi_1(x); \cdots ; \lambda_r, \phi_{r+1}(x))$, the extension of $\mathbf{t}$ by $\phi_{r+1}(x)$; this object is prepared to be enlarged to a type of order $r+1$, $(\tilde{\mathbf{t}}; \lambda_{r+1}, \psi_{r+1}(y))$, simply by taking any negative rational number $\lambda_{r+1} \in \mathbb{Q}^-$ and any irreducible monic polynomial $\psi_{r+1}(y) \in \mathbb{F}_{r+1}[y]$.

The representative of a type plays the analogous role in order r to that played by an irreducible polynomial modulo p in order one.

2.2. **Types versus prime ideals. The Basic algorithm.** Recall that we have fixed a monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$.

Definition 2.5. *A type* $\mathbf{t}$ *of order* r *is said to be* f-complete *if* $\omega_{r+1}^{\mathbf{t}}(f) = 1$.

Theorem 2.6 ([HN, Cor.3.8])**.** *Let* $\mathbf{t}$ *be an* f*-complete type of order* r*. Then the* p*-adic factor* $f_{\mathbf{t}}(x)$ *is irreducible in* $\mathbb{Z}_p[x]$*. Moreover, if* $L/\mathbb{Q}_p$ *is the extension generated by* $f_{\mathbf{t}}(x)$*, we have*

$$e(L/\mathbb{Q}_p) = e_1 \cdots e_r, \quad f(L/\mathbb{Q}_p) = f_0 f_1 \cdots f_r.$$

Thus, an f-complete type singles out a unique prime ideal $\mathfrak{p}$ dividing $p\,\mathbb{Z}_K$, whose ramification index and residual degree can be read in the data of $\mathbf{t}$:

$$e(\mathfrak{p}/p) = e_1 \cdots e_r, \quad f(\mathfrak{p}/p) = f_0 f_1 \cdots f_r.$$

The p-adic factorization process of [HN] consists essentially in the construction of a set $\mathbf{T}$ of f-complete types, that faithfully represents $f(x)$. Thus, it can be interpreted as a *Basic algorithm*, to determine the prime ideal decomposition of $p\,\mathbb{Z}_K$. The types are built iteratively by means of Theorem 2.4, and the theory of Newton polygons of higher order. We start with the set $\mathbf{T}_0(f) := \mathbf{t}_0(f)$, that faithfully represents $f(x)$. We extend the non-f-complete types of this set to types of order one, in order to construct a set $\mathbf{T}_1(f)$ that, again, faithfully represents $f(x)$, etc. At each order r, the extension process is carried out by a main loop that performs the following operations.

Main loop of the Basic algorithm. At the input of a non-f-complete type $\mathbf{t}$ of order $r-1$, for which $\omega_r(f) > 0$, and a representative $\phi_r(x)$:

1) Compute the Newton polygon of r-th order, $N_r(f) = S_1 + \cdots + S_t$, with respect to $\mathbf{t}$ and $\phi_r(x)$.
2) For every side S_j of negative slope $\lambda_{r,j} < 0$, compute the residual polynomial of r-th order, $R_{r,j}(f)(y) \in \mathbb{F}_r[y]$, with respect to $\mathbf{t}$, $\phi_r(x)$ and $\lambda_{r,j}$.
3) Factorize this polynomial in $\mathbb{F}_r[y]$:
$$R_{r,j}(f)(y) = \psi_{r,1}(y)^{a_1} \cdots \psi_{r,s}(y)^{a_s}.$$
4) For every factor $\psi_{r,k}(y)$, compute a representative of the type $\mathbf{t}^{j,k} := (\tilde{\mathbf{t}}; \lambda_{r,j}, \psi_{r,k}(y))$.

For those factors $\psi_{r,k}(y)$ with exponent $a_k = 1$, the type $\mathbf{t}^{j,k}$ is complete. For the remaining types we continue the iterative process.

Thus, each non-complete type of order $r-1$ has sprouted several types of order r, which are called *branches* of the input type $\mathbf{t}$. We have a factorization in $\mathbb{Z}_p[x]$:

$$f_{\mathbf{t}}(x) = \prod_{j,k} f_{\mathbf{t}^{j,k}}(x),$$

with $\deg f_{\mathbf{t}^{j,k}} = e_{r,j} f_{r,k} m_r$. Also, $(\omega_{r+1})^{\mathbf{t}^{j,k}}(f) > 0$, for all j, k, and

$$\omega_r^{\mathbf{t}}(f) = \sum_{j,k} e_{r,j} f_{r,k} (\omega_{r+1})^{\mathbf{t}^{j,k}}(f). \tag{2.1}$$

Hence, the invariant $\omega_r^{\mathbf{t}}(f)$ is an upper bound for the number of irreducible factors of $f_{\mathbf{t}}(x)$, and it is a kind of measure of the distance that is left to complete the analysis of the type $\mathbf{t}$ and its branches (or equivalently, to decompose each $f_{\mathbf{t}}^{j,k}(x)$ into a product of irreducible factors). Also, (2.1) shows that, except for the case in which there is only one branch with $e_r = f_r = 1$, the branches are always closer to be f-complete than $\mathbf{t}$.

We denote by $\mathbf{t}_r(f)$ the set of types of order r obtained by aplying the Main loop to all non-f-complete types of $\mathbf{t}_{r-1}(f)$. We denote by $\mathbf{t}_i(f)^{\mathrm{compl}}$ the subset of the f-complete types of $\mathbf{t}_i(f)$, and we define

$$\mathbf{T}_r(f) := \mathbf{t}_r(f) \cup \left(\bigcup_{0\le i<r} \mathbf{t}_i(f)^{\mathrm{compl}} \right).$$

Proposition 2.7 ([HN, §3]). *$\mathbf{T}_r(f)$ faithfully represents $f(x)$.*

To show that the Basic algorithm deserves this name, we have to prove that, after a finite number of enlargements, all types of $\mathbf{t}_r(f)$ will be complete. To this purpose we introduce another variable to measure how far is a type from being complete, that works even in the unibranch case with $e_r = f_r = 1$. This control variable is defined in terms of *higher indices.*

2.3. **Indices of higher order.** The results of this section are extracted from [HN, §4]. Denote

$$\operatorname{ind}(f) := v\left((\mathbb{Z}_K \colon \mathbb{Z}[\theta]) \right),$$

and recall the well-known relationship, $v(\operatorname{disc}(f)) = 2\operatorname{ind}(f) + v(\operatorname{disc}(K))$, between $\operatorname{ind}(f)$, the discriminant of $f(x)$ and the discriminant of K.

Definition 2.8. *Let $N = S_1 + \cdots + S_t$ be a principal polygon, with finite sides ordered by increasing slopes $\lambda_1 < \cdots < \lambda_t < 0$. Denote by $E_i = \ell(S_i)$, $H_i = H(S_i)$, $d_i = d(S_i)$ the respective length, height and degree of each side* [HN, §1.1]. *We define the* index of the polygon N *to be the nonnegative integer*

$$\operatorname{ind}(N) := \sum_{i=1}^{t} \frac{1}{2}(E_iH_i - E_i - H_i + d_i) + \sum_{1\le i<j\le t} E_iH_j.$$

This number is equal to the number of points with integral coordinates that lie below or on the polygon, strictly above the horizontal line that passes through the last point of N and strictly beyond the vertical axis. Hence, $\operatorname{ind}(N) = 0$ if and only if N has a unique side with height $H = 1$, or length $E = 1$.

Definition 2.9. *Let $\mathbf{t}$ be a type of order $r-1$, and let $\phi_r(x)$ be a representative of $\mathbf{t}$. We define its f-*index *to be the nonnegative integer*

$$\operatorname{ind}_{\mathbf{t}}(f) := \operatorname{ind}_{\mathbf{t},\phi_r}(f) := f_0 \cdots f_{r-1} \operatorname{ind}(N_r^-(f)),$$

the Newton polygon of r-th order taken with respect to $\mathbf{t}$ and $\phi_r(x)$.

*We say that $\mathbf{t}$ is f-*maximal *if $\mathbf{t}$ divides $f(x)$ and $\operatorname{ind}_{\mathbf{t}}(f) = 0$.*

For any natural number $r \ge 1$, we define $\operatorname{ind}_r(f) := \sum_{\mathbf{t}\in\mathbf{t}_{r-1}(f)} \operatorname{ind}_{\mathbf{t}}(f)$.

Since the Newton polygon $N_r^-(f)$ depends on the choice of $\phi_r(x)$, the value of $\operatorname{ind}_{\mathbf{t}}(f)$, and the fact of being f-maximal, depends on this choice too.

Proposition 2.10 ([HN, Lem.4.16]).

a) *If $\mathbf{t}$ is f-complete, then it is f-maximal.*
b) *If $\mathbf{t}$ is f-maximal, then either $\mathbf{t}$ is f-complete, or the output of the Main loop applied to $\mathbf{t}$ is a unique branch of order $r+1$, which is f-complete.*

Thus, the fact that all types of $\mathbf{t}_r(f)$ are complete is essentially equivalent to the fact that they are all maximal. The proof that this will occur after a finite number of iterations is provided by the Theorem of the index.

Theorem 2.11 (Theorem of the index [HN, Thm.4.18])**.** *For all $r \geq 1$,*

$$\operatorname{ind}(f) \geq \operatorname{ind}_1(f) + \cdots + \operatorname{ind}_r(f), \tag{2.2}$$

and equality holds if and only if all types of $\mathbf{t}_r(f)$ are f-maximal.

This theorem shows that after a finite number of iterations all types of $\mathbf{t}_r(f)$ will be f-maximal, because the sum of the right-hand side is bounded by the absolute constant $\operatorname{ind}(f)$. By (b) of Proposition 2.10, either $\mathbf{T}_r(f)$, or $\mathbf{T}_{r+1}(f)$, will contain only f-complete types. By Theorem 2.6 and Proposition 2.7, our family of complete types determines the complete factorization of $p\,\mathbb{Z}_K$ into a product of prime ideals. At this final stage we have necessarily an equality in (2.2), so that we get a computation of $\operatorname{ind}(f)$ as a by-product.

Remark 2.12. If at the end of step 1 of the Main loop of the Basic algorithm, we accumulate to a global variable the value $\operatorname{ind}_{\mathbf{t}}(f)$, the final output of this global variable is $\operatorname{ind}(f)$. In particular, $\operatorname{ind}_{\mathbf{t}}(f)$ is an absolute measure of the distance covered by each iteration of the Main loop, towards the end of the algorithm.

Summing up, we have proved the main theorem of the paper.

Theorem 2.13. *Given a number field K, a generating equation $f(x) \in \mathbb{Z}[x]$, and a prime number p, we can construct a set $\mathbf{T}$ of f-complete types, that faithfully represents $f(x)$. The types of $\mathbf{T}$ are in 1-1 correspondence with the prime ideals of K lying above p, and the ramification index and residual degree of each ideal can be read from data of the corresponding type. Along the construction of $\mathbf{T}$, the algorithm computes the p-valuation of the index of $f(x)$ as well.*

The Theorem of the index and Proposition 2.10 show that the number of iterations of the Main loop is bounded by $\operatorname{ind}(f)$. Actually, in practice, the number of iterations is much lower, because in each step, $\operatorname{ind}_{\mathbf{t}}(f)$ is usually much bigger than one, due to the abundance of the number of points of integer coordinates below an average Newton polygon with a fixed length $\omega_r^{\mathbf{t}}(f)$, and the fact that these points are counted with weight $f_0 \cdots f_{r-1}$.

In the next section we introduce a crucial optimization. A *refinement process* will control at each iteration wether it is strictly necessary to build a type of higher order, or it is possible to keep working in the same order, to avoid an increase of the recursivity in the computations. For instance, the polynomial $f(x) = (x-2)^2 + 2^{2k}$ would require the construction of types of level $\approx k$ in a strict application of the Basic algorithm, while it can be completely analyzed with a refined type of order 1.

3. Optimal representatives of types

3.1. **Detection of optimal representatives.** The construction of types dividing a given polynomial is not canonical: in the construction of the representatives $\phi_r(x)$ one makes some choices, mainly related to lifting certain polynomials over finite fields to polynomials over $\mathbb{Z}$. A natural question arises: are there some choices better than other ones?

Consider the following trivial example: let $p = 2$, $f(x) = x^2 - 4x + 12$, and $K = \mathbb{Q}(\theta) = \mathbb{Q}(\sqrt{-2})$, with $\mathbb{Z}_K = \mathbb{Z}[\sqrt{-2}]$. The polynomial $f(y)$ has only one irreducible factor, $\psi_0(y) = y$, modulo 2; thus, the type of order zero $\mathbf{t} = \psi_0(y)$ gives no information about the factorization of $2\,\mathbb{Z}_K$. The more natural lifting of ψ_0 to $\mathbb{Z}[x]$ is $\phi_1(x) = x$, and the corresponding Newton polygon and residual polynomial determine a unique extension of $\mathbf{t}$ to a type of order one, $(x; -1, y+1)$, which is still not complete, so that we must construct a type of (at least) order 2 to determine the factorization of $2\,\mathbb{Z}_K$. If we choose instead, $\phi_1(x) = x - 2$, we find $f(x) = (x-2)^2 + 2^3$, and the unique extension of $\mathbf{t}$ to a type of order one, $(\phi_1(x); -3/2, y+1)$, is complete. Thus, it is clear that this second choice of $\phi_1(x)$ is better.

While it seems very difficult to predict a priori whether a choice of $\phi_r(x)$ is better than another, it is possible a posteriori to know if our choice was optimal and, if this is not the case, to improve its quality.

Theorem 3.1. *Let $\mathbf{t}^0 \in \mathbf{t}_{r-1}(f)$ be a type of order $r-1$, which is not f-complete, and let $\phi_r(x)$ be a representative of $\mathbf{t}^0$. Let $\mathbf{t} = (\tilde{\mathbf{t}}^0; \lambda_r, \psi_r(y)) \in \mathbf{t}_r(f)$ be one of the branches of $\mathbf{t}^0$, and let $f_{\mathbf{t}}(x) \in \mathbb{Z}_p[x]$ be the factor of $f(x)$ determined by $\mathbf{t}$. Let $\phi'_r(x) \in \mathbb{Z}[X]$ be another representative of $\mathbf{t}^0$. If $e_rf_r > 1$, then,*

a) *The Newton polygon $N'_r(f_{\mathbf{t}})$, with respect to $\mathbf{t}^0$ and $\phi'_r(x)$, is one-sided with slope $\lambda'_r \ge \lambda_r$, and it has the same end point than $N_r(f_{\mathbf{t}})$.*
b) *The residual polynomial $R'_r(f_{\mathbf{t}})(y)$, with respect to $\mathbf{t}^0$, $\phi'_r(x)$ and λ'_r, has only one irreducible factor in $\mathbb{F}_r[y]$; that is, $R'_r(f_{\mathbf{t}})(y) \sim \psi'_r(y)^{a'_r}$, for some monic irreducible polynomial $\psi'_r(y)$.*
c) *Let $(\tilde{\mathbf{t}}^0)' = (\phi_1(x); \cdots; \lambda_{r-1}, \phi'_r(x))$ be the extension of $\mathbf{t}^0$ determined by the choice of $\phi'_r(x)$, and let $\mathbf{t}' = ((\tilde{\mathbf{t}}^0)'; \lambda'_r, \psi'_r(y))$. If $\lambda'_r > \lambda_r$, then $e'_r = f'_r = 1$. If $\lambda'_r = \lambda_r$, then $e'_r = e_r$, $f'_r = f_r$, and $\omega^{\mathbf{t}'}_{r+1}(f) = \omega^{\mathbf{t}}_{r+1}(f)$.*

Therefore, if $e_rf_r > 1$, the representative $\phi_r(x)$ is optimal for this branch $\mathbf{t}$ of order r. The absolute measures $\operatorname{ind}_{\mathbf{t}^0,\phi'_r}(f)$, $\operatorname{ind}_{\mathbf{t}^0,\phi_r}(f)$ are not the right invariant to compare because they incorporate the influence of other branches. If we center our attention on the branch $\mathbf{t}$, there are two situations in which the choice of $\phi'_r(x)$ would lead us to be closer to the end of the analysis of this branch:

(1) $\mathbf{t}$ is replaced by several branches of order r,
(2) $\mathbf{t}$ is replaced by $\mathbf{t}'$, with $\omega^{\mathbf{t}'}_{r+1}(f) < \omega^{\mathbf{t}}_{r+1}(f)$.

Items a), b) show that any choice of $\phi'_r(x)$ leads to replacing the branch $\mathbf{t}$ of $\mathbf{t}^0$, by a single branch $\mathbf{t}'$. Also, if $\lambda'_r = \lambda_r$, we have $\omega^{\mathbf{t}'}_{r+1}(f) = \omega^{\mathbf{t}}_{r+1}(f)$; thus, replacing the type $\mathbf{t}$ by $\mathbf{t}'$ makes no difference at all in this case.

However, if $|\lambda'_r| < |\lambda_r|$, we get a definitely worse approximation to the final solution, because $\omega^{\mathbf{t}}_{r+1}(f) = \omega^{\mathbf{t}'}_r(f)/(e_rf_r)$, by (2.1). Thus, the type $\mathbf{t}$ is much nearer to be complete than $\mathbf{t}'$. Also, if $f_r > 1$, $\operatorname{ind}_{\mathbf{t}}(f)$ will be probably bigger than $\operatorname{ind}_{\mathbf{t}'}(f)$, because each point of integer coordinates below $N^-_{r+1}(f)$ will contribute with a higher weight, $f_0 \cdots f_r$, to the f-index.

Note that a choice of the representative $\phi_r(x)$ of $\mathbf{t}^0$ can be optimal for some branches $\mathbf{t}$ and non-optimal for other branches. We shall see later that the condition $e_rf_r > 1$ is also necessary for the optimality of $\phi_r(x)$ with respect to the branch $\mathbf{t}$.

For the proof of Theorem 3.1, we need an auxiliary result. Fix a type $\mathbf{t}^0$ of order $r-1$ and dividing $f(x)$. For any $\mathbf{n} = (n_0, \dots, n_{r-1}) \in \mathbb{N}^r$, denote $\Phi(\mathbf{n})(x) = p^{n_0}\phi_1(x)^{n_1} \dots \phi_{r-1}(x)^{n_{r-1}}$. Let $\theta \in \overline{\mathbb{Q}}_p$ be a root of $f_{\mathbf{t}^0}(x)$, and $L = \mathbb{Q}_p(\theta)$. In [HN, (27)], an embedding $\mathbb{F}_r \hookrightarrow \mathbb{F}_L$, is defined by

$$\iota_\theta \colon \mathbb{F}_r \hookrightarrow \mathbb{F}_L, \quad z_0 \mapsto \bar{\theta},\ z_1 \mapsto \overline{\gamma_1(\theta)}, \dots,\ z_{r-1} \mapsto \overline{\gamma_{r-1}(\theta)}, \tag{3.1}$$

for certain rational functions $\gamma_i(x) \in \mathbb{Z}(x)$ such that $v(\gamma_i(\theta)) = 0$ [HN, Def.2.13,Cor.3.2].

Lemma 3.2. *Let $\mathbf{t}^0$, θ, L be as above. Let $M(x) \in \mathbb{Z}[x]$ be a polynomial of degree less than m_r. Suppose that $\mathbf{n} \in \mathbb{N}^r$ satisfies $v(M(\theta)) = v(\Phi(\mathbf{n})(\theta))$. Then, the nonzero element $\overline{M(\theta)/\Phi(\mathbf{n})(\theta)} \in \mathbb{F}^*_L$ belongs to $\iota_\theta(\mathbb{F}_r)$, and the element $\iota_\theta^{-1}(\overline{M(\theta)/\Phi(\mathbf{n})(\theta)}) \in \mathbb{F}^*_r$ is independent of the choice of θ.*

Proof. Let $J := \{\mathbf{j} = (j_0, \dots, j_{r-1}) \in \mathbb{N}^r \mid 0 \le j_i < e_if_i,\ \text{for } 0 \le i < r\}$, where we take $e_0 = 1$ by convention. Since $\deg M < m_r$, this polynomial can be written in a unique way as

$$M(x) = \sum_{\mathbf{j}=(j_0,\dots,j_{r-1}) \in J} a_{\mathbf{j}} x^{j_0} \Phi(0, j_1, \dots, j_{r-1})(x),$$

for certain integers $a_{\mathbf{j}}$. By [HN, Lem.4.21], we have

$$v(a_{\mathbf{j}}) \ge \delta_{\mathbf{j}} := v(M(\theta)) - v(\Phi(0, j_1, \dots, j_{r-1})(\theta)),$$

for all $\mathbf{j} \in J$. Let $J_0 = \{\mathbf{j} \in J \mid v(a_{\mathbf{j}}) = \delta_{\mathbf{j}}\}$. Denote $b_{\mathbf{j}} = a_{\mathbf{j}}/p^{\delta_{\mathbf{j}}}$, and $\mathbf{j}' = (\delta_{\mathbf{j}}, j_1, \dots, j_{r-1})$. We can write $M(x)$ as

$$M(x) = \sum_{\mathbf{j} \in J_0} b_{\mathbf{j}} x^{j_0} \Phi(\mathbf{j}')(x) + N(x),$$

where $N(x) \in \mathbb{Z}[x]$ satisfies $v(N(\theta)) > v(M(\theta))$. Now,

$$\frac{M(x)}{\Phi(\mathbf{n})(x)} = \sum_{\mathbf{j} \in J_0} b_{\mathbf{j}} x^{j_0} \Phi(\mathbf{j}' - \mathbf{n})(x) + \frac{N(x)}{\Phi(\mathbf{n})(x)}.$$

By hypothesis, $v(\Phi(\mathbf{j}' - \mathbf{n})(\theta)) = 0$. Since $\omega_{r+1}(\Phi(\mathbf{j}' - \mathbf{n})) = 0$ [HN, Prop.2.15], we have $v_r(\Phi(\mathbf{j}' - \mathbf{n})(x)) = 0$, by [HN, Prop.2.9]. By [HN, Lem.2.16], there exists a sequence $i_1, \dots, i_{r-1}$ of integers, that depend only on $\mathbf{j}'$ and $\mathbf{n}$, such that

$$\Phi(\mathbf{j}' - \mathbf{n})(x) = \gamma_1(x)^{i_1} \cdots \gamma_{r-1}(x)^{i_{r-1}}.$$

Hence, the element of $\mathbb{F}_L^*$,

$$\overline{M(\theta)/\Phi(\mathbf{n})(\theta)} = \sum_{\mathbf{j} \in J_0} \bar{b}_{\mathbf{j}} \bar{\theta}^{j_0} \overline{\Phi(\mathbf{j}' - \mathbf{n})(\theta)},$$

belongs to $\iota_\theta(\mathbb{F}_r)$. Since all the ingredients $a_{\mathbf{j}}$, $\delta_{\mathbf{j}}$, $i_1, \dots, i_{r-1}$ etc. depend only on $\mathbf{t}^0$, the element $\iota_\theta^{-1}(\overline{M(\theta)/\Pi(\theta)}) \in \mathbb{F}_r^*$ is independent of θ. □

Proof of Theorem 3.1. Let $\theta \in \overline{\mathbb{Q}}_p$ be now a root of $f_{\mathbf{t}}(x)$, and $L = \mathbb{Q}_p(\theta)$. Let us prove first that $v(\phi_r(\theta)) \ge v(\phi'_r(\theta))$. In fact, let us show that $v(\phi_r(\theta)) < v(\phi'_r(\theta))$ implies $e_r = f_r = 1$. Let us write $\phi'_r(x) = \phi_r(x) + M(x)$, for certain polynomial $M(x) \in \mathbb{Z}[x]$, of degree less than m_r. If $v(\phi_r(\theta)) < v(\phi'_r(\theta))$, then $v(M(\theta)) = v(\phi_r(\theta)) = (v_r(\phi_r) + |\lambda_r|)/e_1 \cdots e_{r-1}$, by the Theorem of the polygon [HN, Thm.3.1]. Since $\deg M < m_r$ we have $\omega_{r+1}(M) = 0$ [HN, Lem.2.2], and [HN, Prop.2.9] shows that $v_r(M) = e_1 \cdots e_{r-1} v(M(\theta)) = v_r(\phi_r) + |\lambda_r|$; hence λ_r is an integer, and $e_r = 1$.

We use now some other rational functions introduced in [HN, Def.2.13], and the identity $v_r(\phi_r) = e_{r-1} f_{r-1} v_r(\phi_{r-1})$ [HN, Thm.2.11]:

$$\gamma_r(x) = \frac{\Phi_r(x)}{\pi_r(x)^{h_r}} = \frac{\phi_r(x)}{\pi_r(x)^{h_r} \pi_{r-1}(x)^{v_r(\phi_r)/e_{r-1}}}.$$

Denote $\Pi(x) = \pi_r(x)^{h_r} \pi_{r-1}(x)^{v_r(\phi_r)/e_{r-1}}$. Since $v(\gamma_r(\theta)) = 0$, we have $v(\Pi(\theta)) = v(\phi_r(\theta)) = v(M(\theta))$. By [HN, (17)], we can write $\Pi(x) = \Phi(\mathbf{n})(x)$, for some $\mathbf{n} \in \mathbb{N}^r$ that depends only on $\mathbf{t}^0$. Now, if we reduce modulo $\mathfrak{m}_L$ the identity

$$\frac{\phi'_r(\theta)}{\Pi(\theta))} = \gamma_r(\theta) + \frac{M(\theta)}{\Pi(\theta))},$$

Lemma 3.2 shows that $\overline{\gamma_r(\theta)} = -\overline{M(\theta)/\Pi(\theta))}$ belongs to $\iota_\theta(\mathbb{F}_r)$. Since $\overline{\gamma_r(\theta)}$ is a root of $\iota_\theta(\psi_r(y))$ [HN, Prop.3.5], we get $f_r = 1$.

We prove now a) of the theorem. If we show that $v(\phi'_r(\theta))$ takes the same value for all the roots θ of $f_{\mathbf{t}}(x)$, then, by the Theorem of the polygon, $N'_r(f_{\mathbf{t}})$ will be one-sided with slope

$$\begin{aligned} \lambda'_r = v_r(\phi'_r) - e_1 \cdots e_{r-1} v(\phi'_r(\theta)) &= v_r(\phi_r) - e_1 \cdots e_{r-1} v(\phi'_r(\theta)) \\ &\ge v_r(\phi_r) - e_1 \cdots e_{r-1} v(\phi_r(\theta)) = \lambda_r. \end{aligned}$$

Now, if $v(\phi'_r(\theta)) = v(\phi_r(\theta))$ for all θ, then the value $v(\phi'_r(\theta))$ is constant, because the value $v(\phi_r(\theta))$ is constant. Note that $v(M(\theta)) = v_r(M)/e_1 \cdots e_{r-1}$ is independent of θ. If there is one θ_0 with $v(\phi'_r(\theta_0)) < v(\phi_r(\theta_0))$, then $v(M(\theta_0)) = v(\phi'_r(\theta_0)) < v(\phi_r(\theta_0))$. Hence, $v(M(\theta)) < v(\phi_r(\theta))$ for all θ, because both expressions are independent of θ. Thus,

$v(\phi'_r(\theta))=v(M(\theta))$ is constant too. Finally, the polygons $N_r(f_{\mathbf{t}})$, $N'_r(f_{\mathbf{t}})$, have both end point $(\omega_r^{\mathbf{t}^0}(f),v_r^{\mathbf{t}^0}(f))$.

We prove items b), c) simultaneously. Suppose first that $\lambda'_r>\lambda_r$; then, the Theorem of the polygon shows that $v(\phi'_r(\theta))<v(\phi_r(\theta))$. Arguing as above, this implies $e'_r=1$ and $\overline{\gamma'_r(\theta)}\in\iota_\theta(\mathbb{F}_r)$, with $\eta:=\iota_\theta^{-1}(\overline{\gamma'_r(\theta)})\in\mathbb{F}_r^*$ independent of θ. By the Theorem of the residual polynomial [HN, Thm.3.7], if θ runs on all the roots of $f_{\mathbf{t}}(x)$, then $\overline{\gamma'_r(\theta)}$ runs on all the roots of the irreducible factors of $R'_r(f_{\mathbf{t}})(y)$. Hence, $R'_r(f_{\mathbf{t}})(y)\sim(y-\eta)^{a'_r}$, and $f'_r=1$.

Suppose now $\lambda'_r=\lambda_r$, so that $v(\phi_r(\theta))=v(\phi'_r(\theta))$, by the Theorem of the polygon. We distinguish two cases. If $v(M(\theta))=v(\phi_r(\theta))$, arguing as above we get $e_r=1$ and $\overline{\gamma_r(\theta)}=\overline{\gamma'_r(\theta)}+\iota_\theta(\eta)$, for some $\eta\in\mathbb{F}_r^*$ which is independent of θ. In this case, $R'_r(f_{\mathbf{t}})(y)\sim R_r(f_{\mathbf{t}})(y+\eta)$ is a nonzero constant times the power of an irreducible polynomial of degree $f'_r=f_r$. If $v(M(\theta))>v(\phi_r(\theta))$, then $\phi_r(\theta)^{e_r}=\phi'_r(\theta)^{e_r}+N(x)$, where $v(N(\theta))>v(\phi_r(\theta)^{e_r})$. Arguing as above, we get $\overline{\gamma_r(\theta)}=\overline{\gamma'_r(\theta)}$, and this implies $R'_r(f_{\mathbf{t}})(y)\sim R_r(f_{\mathbf{t}})(y)$ and $f'_r=f_r$. This implies $\omega_{r+1}^{\mathbf{t}'}(f)=\omega_{r+1}^{\mathbf{t}}(f)$ too, because $e_rf_r\omega_{r+1}^{\mathbf{t}}(f)=e'_rf'_r\omega_{r+1}^{\mathbf{t}'}(f)$ by (2.1). ☐

3.2. **The process of refinement.** What can be said when $e_r=f_r=1$? In this case, we enlarge the type $\mathbf{t}^0$ to an order r type $\mathbf{t}=(\tilde{\mathbf{t}}^0;-h_r,y-\eta)$, and we find a representative $\phi_{r+1}(x)$ of $\mathbf{t}$, of degree $m_{r+1}=m_r$. Let us emphasize a crucial observation.

Remark 3.3. The polynomial $\phi'_r(x):=\phi_{r+1}(x)$ can be taken too as a representative of $\mathbf{t}^0$.

In fact, $\phi'_r(x)$ has type $\mathbf{t}^0$, and $\omega_r(\phi'_r)=\deg\phi'_r/m_r=1$. We shall show that $\phi'_r(x)$ is always a better representative of $\mathbf{t}^0$ than $\phi_r(x)$; thus, in this case $\phi_r(x)$ is never optimal.

The comparison betwen these two representatives is done by means of the following affine transformation:

$$\mathcal{H}:\mathbb{R}^2\longrightarrow\mathbb{R}^2,\qquad \mathcal{H}(x,y)=(x,y-h_rx).$$

Note that the vertical lines of the plane are invariant under this transformation, and $\mathcal{H}$ acts as a translation on them. Also, $\mathcal{H}$ preserves points of integer coordinates. If S is a side of negative slope, of length ℓ, slope λ and degree d, then $\mathcal{H}(S)$ is a side of length ℓ, slope $\lambda-h_r$ and degree d.

Definition 3.4. *Let h be a positive integer, $\mathbf{t}$ a type of order $r-1$, $\phi_r(x)$ a representative of $\mathbf{t}$, and $P(x)\in\mathbb{Z}[x]$ a nonzero polynomial that is not divisible by $\phi_r(x)$.*

If N is a principal polygon, we denote by N^h the part of N formed by the sides of slope less than $-h$. We define

$$\operatorname{ind}_{\mathbf{t}}^h(P):=\operatorname{ind}_{\mathbf{t},\phi_r}^h(P):=f_0\cdots f_{r-1}\left(\operatorname{ind}(N_r^h(P))-\frac{1}{2}h\ell(\ell-1)\right),\tag{3.2}$$

where $\ell=\ell(N_r^h(P))$, and the Newton polygon is taken with respect to $\mathbf{t}$ and $\phi_r(x)$.

This number $\operatorname{ind}_{\mathbf{t}}^h(P)$ is equal to $f_0\cdots f_{r-1}$ times the number of points of integer coordinates in the region of the plane determined by the points that lie below (or on) $N_r^h(P)$, strictly above the line L_{-h} of slope $-h$ that passes through the last point of the polygon, and strictly beyond the vertical axis. The term $h\ell(\ell-1)/2$ takes care of the points of integer coordinates in the triangle determined by L_{-h}, the vertical axis and the horizontal line that passes through the last point of $N_r^h(P)$.

Let us introduce some notation. Let $N_{r+1}(-)$, denote the Newton polygon with respect to $\mathbf{t}$ and $\phi_{r+1}(x)$. Let $N'_r(-)$ denote the Newton polygon with respect to $\mathbf{t}^0$ and $\phi'_r(x)=\phi_{r+1}(x)$. For any negative rational number λ, let $R'_\lambda(-)(y)\in\mathbb{F}_r[y]$ denote the residual polynomial in order r, with respect to $\mathbf{t}^0$, $\phi'_r(x)$, λ, and let $R_\lambda(-)(y)\in\mathbb{F}_r[y]$ denote the residual polynomial in order $r+1$, with respect to $\mathbf{t}$, $\phi_{r+1}(x)$, λ.

Proposition 3.5. *Let $\mathbf{t}^0\in\mathbf{t}_{r-1}(f)$ be a non-complete type of order $r-1$, and let $\mathbf{t}=(\tilde{\mathbf{t}}^0;-h_r,y-\eta)\in\mathbf{t}_r(f)$ be a branch of $\mathbf{t}^0$ such that $e_r=f_r=1$. Let $\phi_{r+1}(x)$ be a representative* *of* $\mathbf{t}$*, and let* $\phi'_r(x)=\phi_{r+1}(x)$ *be the same polynomial, considered as a representative of* $\mathbf{t}^0$*. Let* $P(x)\in\mathbb{Z}_p[x]$ *be a nonzero polynomial. Then, with the previous notations,*

a) $(N')_r^{h_r}(P)=\mathcal{H}(N_{r+1}^-(P))$,

b) $\operatorname{ind}_{\mathbf{t}}(P)=\operatorname{ind}^h_{\mathbf{t}^0,\phi'_r}(P)$,

c) *There exists a nonzero constant* $\epsilon\in\mathbb{F}_r$ *that depends only on* $\mathbf{t}^0$*, such that, for any* $\lambda=-h/e$*, with* h,e *positive coprime integers, we have* $R_\lambda(P)(y)=\epsilon^sR'_{\lambda-h_r}(P)(\epsilon^ey)$*, where* s *is the initial abscissa of the* λ*-component of* $N_{r+1}^-(P)$ [HN, §1.1].

Proof. We shall denote by v_{r+1} the p-adic valuation attached to $\mathbf{t}$, and by v_r the p-adic valuation attached to $\mathbf{t}^0$. Consider the ϕ_{r+1}-adic development of $P(x)$, which is simultaneously its ϕ'_r-adic development:

$$P(x)=\sum_{0\le i}a_i(x)\phi_{r+1}(x)^i=\sum_{0\le i}a_i(x)\phi'_r(x)^i.$$

For any $0\le i$, denote $u_i=v_{r+1}(a_i\phi_{r+1}^i)$, $u'_i=v_r(a_i(\phi'_r)^i)$, so that the points (i,u_i) determine the Newton polygon $N_{r+1}(P)$, and the points (i,u'_i) determine the Newton polygon $N'_r(P)$. Since $\deg a_i<m_r=m_{r+1}$,

$$v_r(a_i)=e_1\cdots e_{r-1}v(a_i(\theta))=e_1\cdots e_{r-1}e_rv(a_i(\theta))=v_{r+1}(a_i),$$

where θ is any root of $f_{\mathbf{t}}(x)$ in $\overline{\mathbb{Q}}_p$. By [HN, Prop.2.7+Thm.2.11],

$$v_{r+1}(\phi_{r+1})=f_re_rv_{r+1}(\phi_r)=v_{r+1}(\phi_r)=e_rv_r(\phi_r)+h_r=v_r(\phi_r)+h_r=v_r(\phi'_r)+h_r.$$

This proves a), because $u_i=v_{r+1}(a_i\phi_{r+1}^i)=v_r(a_i(\phi'_r)^i)+ih_r=u'_i+ih_r$.

Item b) is an immediate consequence, because $\mathcal{H}$ transforms the horizontal line that passes through the last point of $N_{r+1}^-(P)$ into the line L_{-h_r} of slope $-h_r$ that passes through the last point of $N_r^h(P)$ (cf. the figure below).

Let us prove c). The definition of the residual coefficients and the residual polynomials is given in [HN, Defs.2.20-2.21]. Denote $N'=(N')_r^{h_r}(P)$. To every integer abscissa, $0\le i\le\ell(N')$, one attaches a residual coefficient c_i of $N_{r+1}^-(P)$, and a residual coefficient c'_i of N', given by

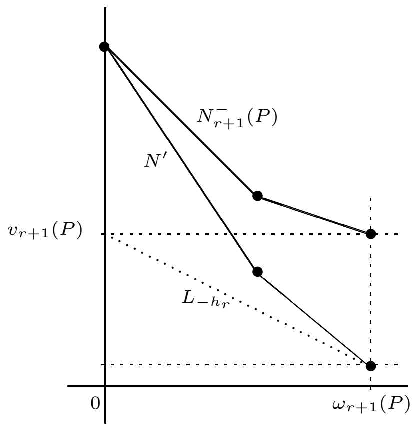
$$c_i=\begin{cases} z_r^{t_r(i)}R_r(a_i)(z_r), & \text{if } (i,u_i) \text{ lies on } N_{r+1}^-(P),\\ 0, & \text{otherwise.}\end{cases}$$

$$c'_i=\begin{cases} z_{r-1}^{t'_{r-1}(i)}R_{r-1}(a_i)(z_{r-1}), & \text{if } (i,u'_i) \text{ lies on } N',\\ 0, & \text{otherwise.}\end{cases}$$

By a), the points (i,u_i), (i,u'_i), lying on the respective polygons have the same abscissas. Suppose that i is such an abscissa. For $j=r,r-1$, denote by $s_j(a_i)$ the initial abscissa of the λ_j-component of $N_j(a_i)$ [HN, §1.1]. Since $e_r=1$, we can choose $\ell_r=0$. Since $\deg(a_i)<m_r$, the polygon $N_r(a_i)$ is reduced to the point $(0,v_r(a_i))$. This implies that $t_r(i)=(s_r(a_i)-\ell_ru_i)/e_r=0$; also, $R_r(a_i)(y)$ is a constant, equal to $z_{r-1}^{t_{r-1}(0)}R_{r-1}(a_i)(z_{r-1})$. The exponents $t'_{r-1}(i)$, and $t_{r-1}(0)$ are given by

$$t'_{r-1}(i)=(s_{r-1}(a_i)-\ell_{r-1}v_r(a_i(\phi'_r)^i))/e_{r-1},\ t_{r-1}(0)=(s_{r-1}(a_i)-\ell_{r-1}v_r(a_i))/e_{r-1}.$$

Hence, $c_i=\epsilon^ic'_i$, where $\epsilon=(z_{r-1})^{\ell_{r-1}v_r(\phi_r)/e_{r-1}}$. Since $R_\lambda(P)(y)=c_s+c_{s+e}y+\cdots+c_{s+de}y^d$, $R'_{\lambda-h_r}(P)(y)=c'_s+c'_{s+e}y+\cdots+c'_{s+de}y^d$, we get $R_\lambda(P)(y)=\epsilon^sR'_{\lambda-h_r}(P)(\epsilon^ey)$. □



Proposition 3.5, applied to $P(x) = f(x)$, shows that $\phi'_r(x)$ is a better representative of $\mathbf{t}^0$ than $\phi_r(x)$, in what the analysis of the branch $\mathbf{t}$ concerns. Actually, we have proved something stronger: the information obtained by applying to $\mathbf{t}$ the Basic algorithm in order $r+1$, is exactly the same information obtained by applying the Basic algorithm to $\mathbf{t}^0$ in order r, as long as we take $\phi'_r(x)$ as a representative, we analyze $(N')^{h_r}_r(f)$ instead of the whole $(N')^-_r(f)$, and we replace $\operatorname{ind}_{\mathbf{t}}(f)$ by $\operatorname{ind}^{h_r}_{\mathbf{t}^0,\phi'_r}(f)$. By "to obtain the same information" we mean to obtain the same number of new branches, the same (decreased) value of $\omega_{r+1}(f)$ for each of them, and to cover the same distance to the end of the analysis of the branch $\mathbf{t}$.

Moreover, let $\lambda \in \mathbb{Q}^-$ be the slope of a side of $N^-_{r+1}(f)$, $\psi(y)$ a monic irreducible factor of $R_\lambda(f)(y)$, and $\mathbf{t}'' = (\tilde{\mathbf{t}}^0; -h_r, \phi_{r+1}; \lambda, \psi(y))$ the corresponding branch of $\mathbf{t}$ of order $r+1$. By Proposition 3.5, this branch mirrors a branch of order r, $\mathbf{t}' = ((\tilde{\mathbf{t}}^0)'; \lambda - h_r, c^{\deg\psi}\psi(c^{-1}y))$, of $\mathbf{t}^0$, with respect to the choice of $\phi'_r(x)$ as a representative.

Corollary 3.6. *The types $\mathbf{t}'$ and $\mathbf{t}''$ are f-equivalent.*

Proof. If $\lambda = h_{r+1}/e_{r+1}$, with h_{r+1}, e_{r+1} positive coprime integers, then $\lambda - h_r = (h_{r+1} - e_{r+1}h_r)/e_{r+1}$, with coprime numerator and denominator; thus, $e^{\mathbf{t}''}_{r+1} = e^{\mathbf{t}'}_r = e_{r+1}$. Also, $f^{\mathbf{t}''}_{r+1} = f^{\mathbf{t}'}_r = \deg\psi$, so that $m^{\mathbf{t}''}_{r+2} = e_{r+1}\deg\psi\deg\phi_{r+1} = m^{\mathbf{t}'}_{r+1}$. By c) of Proposition 3.5, applied to $P(x) = f_{\mathbf{t}'}(x)$, we have $\omega^{\mathbf{t}''}_{r+2}(f_{\mathbf{t}'}) = \omega^{\mathbf{t}'}_{r+1}(f_{\mathbf{t}'})$, and

$$\deg f_{\mathbf{t}'} = m^{\mathbf{t}'}_{r+1}\omega^{\mathbf{t}'}_{r+1}(f_{\mathbf{t}'}) = m^{\mathbf{t}''}_{r+2}\omega^{\mathbf{t}''}_{r+2}(f_{\mathbf{t}'}).$$

This shows that $f_{\mathbf{t}'}(x)$ has type $\mathbf{t}''$; since $\deg f_{\mathbf{t}'} = \deg f_{\mathbf{t}''}$, we have necessarily $f_{\mathbf{t}'} = f_{\mathbf{t}''}$. □

This observation leads to an important optimization of the Basic algorithm. Whenever we apply the Main loop to a type $\mathbf{t}^0$ and one of the outputs is a non-complete branch $\mathbf{t} = (\tilde{\mathbf{t}}_0; -h_r, y - \eta)$, with $e_r = f_r = 1$:

(1) we replace $\mathbf{t}$ by the type $\mathbf{t}^0$ itself, but taking $\phi'_r(x) = \phi_{r+1}(x)$ as a new representative,
(2) we store the cutting slope h_r as one of the data of the last level of $\mathbf{t}^0$, so that when the turn comes to apply the Main loop to the new $\mathbf{t}^0$, only the sides of slope less than $-h_r$ of $(N')^-_r(f)$ will be analyzed.

We call this a *refinement step*, and we use the term *Montes' algorithm* to name the algorithm that is obtained from the Basic algorithm by applying a refinement step to every branch with $e_rf_r = 1$. By Corollary 3.6, every future branch $\mathbf{t}'$ of $\mathbf{t}$ is replaced by an equivalent branch $\mathbf{t}''$ of the new $\mathbf{t}^0$, so that the two algorithms are equivalent. By Proposition 3.5, the distance to the end of the algorithm covered by one application of the Main loop of Montes' algorithm is measured by $\operatorname{ind}^{h_r}_{\mathbf{t}^0,\phi'_r}(f)$.

However, the refinement steps cause a strong diminution of the complexity. In fact, passing from order r to order $r+1$ introduces a new level of recursivity in the basic tasks of the Main loop. Therefore, in Montes' algorithm the same information is obtained by working in a lower order. For instance, if in the Basic algorithm we find a branch of order $r+n$, with n successive levels with $e_{r+i}f_{r+i}=1$, for $0\le i<n$,

$$\tilde{\mathbf{t}}=(\phi_1(x);\lambda_1,\phi_2(x);\cdots;\lambda_{r-1},\phi_r(x);-h_r,\phi_{r+1}(x);\cdots;-h_{r+n-1},\phi_{r+n}(x)).$$

Starting with the trunk $(\phi_1(x);\lambda_1,\phi_2(x);\cdots;\lambda_{r-1},\phi_r(x))$, we reached $\tilde{\mathbf{t}}$ by applying the Main loop n times in orders $r,r+1,\dots,r+n$. If we refine, this type collapses to

$$\tilde{\mathbf{t}}'=(\phi_1(x);\lambda_1,\phi_2(x);\cdots;\lambda_{r-1},\phi'_r(x)),$$

with $\phi'_r(x)=\phi_{r+n}(x)$, and we reach $\tilde{\mathbf{t}}'$ by applying the Main loop n times too, but always in order r.

In order to homogenize the flow of the algorithm, we introduce a variable H_r that stores an integer value at each r-th level of each type. Initially, it is given the value zero, and it is changed to $H_r=h_r$ if we fall in a refinement step. This allows us to use a general Main loop, which presents only two differences with respect to the Main loop of the Basic algorithm:

- in step 1 only the sides of slope less than $-H_r$ of the Newton polygon of $f(x)$ are analyzed,
- in step 4, after computing the representative ϕ_{r+1} of a non-complete new branch of order r, we proceed as follows: if $e_rf_r>1$, we take the new branch as one of the output branches, with the value $H_{r+1}=0$. If $e_rf_r=1$, we take the input type as one of the output branches, with $H_r=h_r$, and $\phi'_r=\phi_{r+1}$ as a new representative.

We can interpret also the refinement steps as a search for the optimal representatives. The search is performed by applying the Basic algorithm and not enlarging the types till a branch with $e_rf_r>1$ is found.

Summing up, Montes' algorithm has the same number of iterations than the basic algorithm, but a much lower complexity. It works only with optimal representatives, and it works always at the minimum order possible till a new optimal representative forces to pass to a higher order.

In spite of this apparent strong optimality, one could speculate on an improvement based on a more intelligent way to pass from an optimal type of order r to an optimal type of order $r+1$. The search for an optimal representative is done by blind lifting of certain polynomials over finite fields to $\mathbb{Z}$, and a blind application of the Basic algorithm (without raising the order). This is extremely fast in practice, but there could exist a more direct way to obtain the next optimal representative.

3.3. **Computation of the index with Montes' algorithm.**

Denote by $\mathbf{t}_r^{\mathrm{opt}}(f)$ the set of types of order r that are produced by Montes' algorithm. The sets $\mathbf{t}_r^{\mathrm{opt}}(f)$ are quite different from the sets $\mathbf{t}_r(f)$ produced by the Basic algorithm, which were crucial in the definition of $\operatorname{ind}_r(f)$ and the proof of the Theorem of the index. We need to compare in some sense these two types of sets. This is provided by Proposition 3.8 below, which is similar to Proposition 3.5, but going in the opposite direction.

Let's go back to the situation of Corollary 3.6. Suppose that the Basic algorithm is working with a type $\mathbf{t}^0$ of order $r-1$, and it finds a branch of order $r+1$ (we denote it now by $\mathbf{t}$ instead of $\mathbf{t}''$):

$$\mathbf{t}=(\tilde{\mathbf{t}}^0;-h_r,\phi_{r+1};\lambda_{r+1},\psi_{r+1}(y))\in\mathbf{t}_{r+1}(f),$$

as the result of two consecutive enlargements of $\mathbf{t}_0$, and we have $e_r^{\mathbf{t}}f_r^{\mathbf{t}}=1$. Let $c=\epsilon^{e_{r+1}^{\mathbf{t}}}$ be the constant of c) of Proposition 3.5, and consider the branch of order r of $\mathbf{t}^0$,

$$\mathbf{t}'=((\tilde{\mathbf{t}}^0)';\lambda'_r,\psi'_r(y)),\quad \lambda'_r=\lambda_{r+1}-h_r,\ \psi'_r(y)=c^{f_{r+1}^{\mathbf{t}}}\psi_{r+1}(c^{-1}y).$$

By Corollary 3.6, $\mathbf{t}'$ is f-equivalent to $\mathbf{t}$.

Remark 3.7. Any representative $\phi'(x)$ of $\mathbf{t}'$ can be taken too as a representative of $\mathbf{t}$.

In fact, along the proof of Corollary 3.6 we saw that $m_{r+2}^{\mathbf{t}} = m_{r+1}^{\mathbf{t}'}$, so that $\phi'(x)$ has the right degree; thus, it is sufficient to check that $\omega_{r+2}^{\mathbf{t}}(\phi') = 1$, and this is given by c) of Proposition 3.5 applied to $P(x) = \phi'(x)$.

Let $N_{r+2}(-)$, denote the principal Newton polygon of order $r+2$, with respect to $\mathbf{t}$ and $\phi(x) := \phi'(x)$. Let $(N')_{r+1}(-)$ denote the principal Newton polygon with respect to $\mathbf{t}'$ and $\phi'(x)$. Denote

$$\mathbb{F}_r := \mathbb{F}_r^{\mathbf{t}'} = \mathbb{F}_r^{\mathbf{t}} = \mathbb{F}_{r+1}^{\mathbf{t}}; \quad \mathbb{F} := \mathbb{F}_{r+2}^{\mathbf{t}} = \mathbb{F}_r[y]/\psi_{r+1}(y) = \mathbb{F}_r[y]/\psi_{r+1}(c^{-1}y) = \mathbb{F}_{r+1}^{\mathbf{t}'}.$$

For any $\lambda = -h/e$, with h, e coprime positive integers, let $R_\lambda(-)(y) \in \mathbb{F}[y]$ denote the residual polynomial in order $r+2$, with respect to $\mathbf{t}$, $\phi(x)$, λ, and let $R'_\lambda(-)(y) \in \mathbb{F}[y]$ denote the residual polynomial in order $r+1$, with respect to $\mathbf{t}'$, $\phi'(x)$, λ.

Let us write $e_{r+1} = e_{r+1}^{\mathbf{t}} = e_r^{\mathbf{t}'}$, $h_{r+1} = h_{r+1}^{\mathbf{t}}$. For the type $\mathbf{t}'$ we have $\ell_r^{\mathbf{t}'} h_r^{\mathbf{t}'} - (\ell'_r)^{\mathbf{t}'} e_r^{\mathbf{t}'} = 1$. Since $h_r^{\mathbf{t}'} = h_{r+1} - e_{r+1}h_r$, this can be written as

$$\ell_r^{\mathbf{t}'} h_{r+1} - \left((\ell'_r)_r^{\mathbf{t}'} + \ell_r^{\mathbf{t}'} h_r \right) e_{r+1} = 1.$$

For the type $\mathbf{t}$ we have $\ell_{r+1}^{\mathbf{t}} h_{r+1} - (\ell'_{r+1})^{\mathbf{t}} e_{r+1} = 1$. Therefore, we can choose $\ell_{r+1}^{\mathbf{t}} = \ell_r^{\mathbf{t}'}$, $(\ell'_{r+1})^{\mathbf{t}} = (\ell'_r)^{\mathbf{t}'} + \ell_r^{\mathbf{t}'} h_r$.

Proposition 3.8. *Let $P(x) \in \mathbb{Z}[x]$ be a nonzero polynomial. Suppose we choose $\ell_{r+1}^{\mathbf{t}} = \ell_r^{\mathbf{t}'}$. With the previous notations,*

a) *$N_{r+2}(P) = (N')_{r+1}(P)$.*
b) *$\operatorname{ind}_{\mathbf{t}}(P) = \operatorname{ind}_{\mathbf{t}'}(P)$.*
c) *There exists a constant $\tau \in \mathbb{F}_r^*$, depending only on $\mathbf{t}'$, such that for any $\lambda = -h/e$, with h, e coprime positive integers, we have $R_\lambda(P)(y) = \tau^u R'_\lambda(P)(\tau^{-h}y)$, where u is the ordinate of the initial point of the λ-component of $N_{r+2}(P)$.*

Proof. For any polynomial $a(x) \in \mathbb{Z}[x]$,

$$v_{r+2}^{\mathbf{t}}(a)/e_{r+1}^{\mathbf{t}} = v_{r+2}^{\mathbf{t}}(a)/e_{r+1}, \qquad v_{r+1}^{\mathbf{t}'}(a)/e_r^{\mathbf{t}'} = v_{r+1}^{\mathbf{t}'}(a)/e_{r+1},$$

are the ordinates at the origin of the lines $L_\lambda(N_{r+1}(a))$, $L_{\lambda - h_r}((N')_r(a))$, respectively [HN, Def.2.5]. These two ordinates at the origin coincide by a) of Proposition 3.5, so that $v_{r+2}^{\mathbf{t}} = v_{r+1}^{\mathbf{t}'}$. This proves a), and b) is an immediate consequence.

Consider the ϕ-adic development $P(x) = \sum_{0 \le i} a_i(x)\phi(x)^i$, and denote $u_i = v_{r+1}^{\mathbf{t}'}(a_i\phi^i) = v_{r+2}^{\mathbf{t}}(a_i\phi^i)$, for all $i \ge 0$. Let $\{c_i\}_{i \ge 0}$ be the residual coefficients of $N_{r+2}(P)$, and $\{c'_i\}_{i \ge 0}$ be the residual coefficients of $(N')_{r+1}(P)$. Since the two polygons are constructed from the same set of points (i, u_i) of the plane, we have $c_i = 0$ if and only if $c'_i = 0$. Let i be an integer abscissa such that the point (i, u_i) lies on $N_{r+2}(P) = (N')_{r+1}(P))$, so that $c_i c'_i \ne 0$. In this case, we have by definition,

$$c_i = (z_{r+1})^{t_{r+1}(i)} R_\lambda(a_i)(z_{r+1}), \quad c'_i = (z'_r)^{t'_r(i)} R'_\lambda(a_i)(z_r).$$

By definition, $z_{r+1} \equiv y \pmod{\psi_{r+1}(y)}$, and $z'_r \equiv y \pmod{\psi_{r+1}(c^{-1}y)}$ in $\mathbb{F}_r[y]$; thus, $cz_{r+1} = z'_r$. By a) of Proposition 3.5 applied to $P(x) = a_i(x)$, we have $s_{r+1}^{\mathbf{t}}(a_i) = s_r^{\mathbf{t}'}(a_i)$; since $\ell_{r+1}^{\mathbf{t}} = \ell_r^{\mathbf{t}'}$ by hypothesis, and $e_{r+1}^{\mathbf{t}} = e_r^{\mathbf{t}'}$, we get $t_{r+1}(i) = t'_r(i)$. Therefore, by c) of Proposition 3.5,

$$c'_i/c_i = (z'_r/z_{r+1})^{t'_r(i)} \epsilon^{-s_r^{\mathbf{t}'}(a_i)} = \epsilon^{-\ell_r^{\mathbf{t}'} u_i}.$$

If (s, u) is the initial point of the λ-component of $N_{r+2}(P) = (N')_{r+1}(P)$, and $i = s + je$, we have $u_i = u - jh$, so that $R_\lambda(P)(y) = \tau^u R'_\lambda(P)(\tau^{-h}y)$, for $\tau = \epsilon^{\ell_r^{\mathbf{t}'}}$. □

Therefore, we are able to deduce from the optimal types constructed by Montes' algorithm, relevant information about the general types that would be constructed by the Basic algorithm. In particular, by an alternative and iterative application of Propositions 3.5 and 3.8, all values of $\operatorname{ind}_{\mathbf{t}}(f)$, for all $\mathbf{t}\in\mathbf{t}_r(f)$, can be captured along the flow of Montes' algorithm.

Remark 3.9. If at the end of step 1 of the Main loop of Montes' algorithm, we accumulate to a global variable the value $\operatorname{ind}^{h_r}_{\mathbf{t}^0}(f)$, the final output of this global variable is $\operatorname{ind}(f)$.

In fact, if the input type $\mathbf{t}^0$ is the result of an ordinary enlargement, then $h_r=0$ and $\operatorname{ind}^{h_r}_{\mathbf{t}^0}(f)=\operatorname{ind}_{\mathbf{t}^0}(f)$; if $\mathbf{t}^0$ is the result of a refinement step then, by Proposition 3.5, $\operatorname{ind}^{h_r}_{\mathbf{t}^0}(f)$ is equal to the f-index of the type that the Basic algorithm would have produced if we had not refined. Proposition 3.8 guarantees that the future development of Montes's algorithm after a refinement step, mirrors the future development of the Basic algorithm.

4. Generators of the prime ideals

In this section we compute generators of the prime ideals lying above p in terms of the output of Montes' algorithm: a list $\mathbf{T}=\{\mathbf{t}_{\mathfrak{p}_1},\dots,\mathbf{t}_{\mathfrak{p}_1}\}$, of f-complete types with optimal representatives, which are in $1-1$ correspondence with the prime ideals $\mathfrak{p}_1,\dots,\mathfrak{p}_g$ of K dividing $p\,\mathbb{Z}_K$. We write $e_r^{\mathfrak{p}}$, $\lambda_r^{\mathfrak{p}}$, $\phi_r^{\mathfrak{p}}$, etc. to indicate that a datum corresponds to the type $\mathbf{t}_{\mathfrak{p}}$. Recall that $e(\mathfrak{p}/p)=e_1^{\mathfrak{p}}\cdots e_r^{\mathfrak{p}}$ and $f(\mathfrak{p}/p)=f_0^{\mathfrak{p}}\cdots f_r^{\mathfrak{p}}$, can be read in the data of $\mathbf{t}_{\mathfrak{p}}$. We choose a root $\theta\in\overline{\mathbb{Q}}$ of $f(x)$, and denote by $\theta^{\mathfrak{p}}\in\overline{\mathbb{Q}}_p$ the root of $f_{\mathbf{t}_{\mathfrak{p}}}(x)$, image of θ under the topological embedding $K\hookrightarrow K_{\mathfrak{p}}$.

If $\mathbf{t}\in\mathbf{T}$ has order zero, and $\phi(x)$ is a representative of $\mathbf{t}$, then the corresponding prime ideal is generated by $(p,\phi(x))$ by Kummer's criterion. If $\mathbf{t}\in\mathbf{T}$ has order one and its truncation $\mathbf{t}_0$ of order zero has $\operatorname{ind}_{\mathbf{t}_0}(f)=0$, then the program computes generators of the corresponding prime ideal by using Dedekind's criterion.

From now on, we fix a type $\mathbf{t}=\mathbf{t}_{\mathfrak{p}}$, corresponding to a prime ideal $\mathfrak{p}$ that did not fall in those special cases. We omit the superscript $(\;)^{\mathfrak{p}}$ for the data of $\mathbf{t}$. Let r be the order of $\mathbf{t}$. We want to compute an integral element $\alpha=\alpha_{\mathfrak{p}}\in\mathbb{Z}_K$ satisfying

$$v_{\mathfrak{p}}(\alpha)=1;\quad v_{\mathfrak{q}}(\alpha)=0,\ \forall\mathfrak{q}\mid p,\ \mathfrak{q}\ne\mathfrak{p};\quad v_{\mathfrak{l}}(\alpha)\ge 0,\ \forall\mathfrak{l}\nmid p,$$

so that the ideal $\mathfrak{p}$ is generated by p and α.

Let us first construct an element $\beta=\beta_{\mathfrak{p}}\in K$ such that $v_{\mathfrak{p}}(\beta)=1$. To this end we compute first a representative $\phi_{r+1}(x)$ of $\mathbf{t}$. Since $\mathbf{t}$ is f-complete, $\omega_{r+1}(f)=1$ and $\phi_r(x)\ne f(x)$. The Newton polygon $N_{r+1}(f)$, with respect to $\mathbf{t}$ and $\phi_{r+1}(x)$, is one-sided of length one, and integer slope $-H$, where H is the height of the side. By the theorem of the polygon,

$$\begin{aligned}v_{\mathfrak{p}}(\phi_{r+1}(\theta))&=e(\mathfrak{p}/p)v(\phi_{r+1}(\theta^{\mathfrak{p}}))=v_{r+1}(\phi_{r+1})+H\\&=e_rf_rv_{r+1}(\phi_r)+H=e_rf_r(e_rv_r(\phi_r)+h_r)+H,\end{aligned}$$

the last two equalities by [HN, Thm.2.11,Prop.2.7]. On the other hand, $\omega_{r+1}(\phi_r)=0$ [HN, Prop.2.15], and [HN, Prop.2.9] shows that

$$v_{\mathfrak{p}}(\phi_r(\theta))=e(\mathfrak{p}/p)v(\phi_r(\theta^{\mathfrak{p}}))=e_r(v_r(\phi_r)+(h_r/e_r))=e_rv_r(\phi_r)+h_r.$$

Therefore, if we consider the element $\beta\in K$, defined as

$$\beta:=\frac{\phi_{r+1}(\theta)}{\phi_r(\theta)^{e_rf_r}},$$

we have $v_{\mathfrak{p}}(\beta)=H$. Thus, our aim is to find a kind of "worse possible" representative $\phi_{r+1}(x)$ of $\mathbf{t}$; that is, one satisfying $H=1$. To this end, we compute a blind $\phi_{r+1}(x)$. If $H=1$ we are done; if $H>1$ we use a subroutine based on [HN, Prop.2.10], to construct a polynomial $P(x)\in\mathbb{Z}[x]$ with the following properties:

$$\deg P<m_{r+1},\quad v_{r+1}(P)=v_{r+1}(\phi_{r+1})+1,\quad R_r(P)(y)=1.$$

The point is that $\tilde{\phi}_{r+1} := \phi_{r+1}(x) + P(x)$ is another representative of $\mathbf{t}$, and it has $H = 1$. In fact, $\deg \tilde{\phi}_{r+1} = \deg \phi_{r+1}$ and $\omega_{r+1}(\tilde{\phi}_{r+1}) = \omega_{r+1}(\phi_{r+1}) = 1$, by [HN, Prop.2.8], so that $\tilde{\phi}_{r+1}$ is of type $\mathbf{t}$ and it $\tilde{\phi}_{r+1}$ is a representative of $\mathbf{t}$. Now, since $\deg P < m_{r+1}$, we have

$$v(P(\theta^{\mathfrak{p}})) = \frac{v_{r+1}(P)}{e(\mathfrak{p}/p)} = \frac{v_{r+1}(\phi_{r+1}) + 1}{e(\mathfrak{p}/p)} < \frac{v_{r+1}(\phi_{r+1}) + H}{e(\mathfrak{p}/p)} = v(\phi_{r+1}(\theta^{\mathfrak{p}})).$$

Thus, $v(\tilde{\phi}_{r+1}(\theta^{\mathfrak{p}})) = v(P(\theta^{\mathfrak{p}})) = (v_{r+1}(\phi_{r+1}) + 1)/e(\mathfrak{p}/p)$, and

$$\beta := \frac{\tilde{\phi}_{r+1}(\theta)}{\phi_r(\theta)^{e_r f_r}} \Longrightarrow v_{\mathfrak{p}}(\beta) = 1.$$

Our next step is to compute the values $v_{\mathfrak{q}}(\beta)$, for all other primes $\mathfrak{q}$ lying above p, $\mathfrak{q} \ne \mathfrak{p}$.

Definition 4.1. *We say that $\mathbf{t}_{\mathfrak{q}}$ dominates $\mathbf{t}$, and we write $\mathbf{t}_{\mathfrak{q}} > \mathbf{t}$, if $\mathbf{t}_{\mathfrak{q}}$ is a branch of $(\mathbf{t})_{r-1}$ originated from a side of $N_{\mathbf{t},\phi_r}(f)$ of slope $\lambda < \lambda_r$. In this case we denote $\lambda_{\mathfrak{p}}^{\mathfrak{q}} = \lambda$ and we call it the dominating slope of $\mathbf{t}_{\mathfrak{q}}$ over $\mathbf{t} = \mathbf{t}_{\mathfrak{p}}$.*

Proposition 4.2. *Let $\mathfrak{q}$ be a prime ideal of K lying above p, $\mathfrak{q} \ne \mathfrak{p}$. Let s be the order of $\mathbf{t}_{\mathfrak{q}}$. Then,*

$$v_{\mathfrak{q}}(\beta) = \begin{cases} e_r f_r (e_r^{\mathfrak{q}} \cdots e_s^{\mathfrak{q}})(\lambda_{\mathfrak{p}}^{\mathfrak{q}} - \lambda_r), & \text{if } \mathbf{t}_{\mathfrak{q}} > \mathbf{t}, \\ 0, & \text{otherwise }. \end{cases}$$

Proof. Let r_0 be minimal with the property $(\mathbf{t}_{\mathfrak{q}})_{r_0} \ne \mathbf{t}_{r_0}$. Necessarily $r_0 \le \min\{r, s\}$, because the types $\mathbf{t}$, $\mathbf{t}_{\mathfrak{q}}$ are complete. Let us deal first with the case $r_0 < r$. Since $(\mathbf{t}_{\mathfrak{q}})_{r_0-1} = \mathbf{t}_{r_0-1}$, for some primitive choice $\phi_{r_0}(x)$ of a representative of this type, the Main loop produced (at least) two different branches, that later developed to produce the types $\mathbf{t}$, $\mathbf{t}_{\mathfrak{q}}$. Some of these branches might have been refined, causing a change of this representative at level $r_0 - 1$; however, arguing with these primitive branches if it were necessary, we can assume that $\phi_{r_0} = \phi_{r_0}^{\mathfrak{p}} = \phi_{r_0}^{\mathfrak{q}}$. This might change the values of the data of the r-th level, but this is not relevant in our arguments. After our assumption, $v_i^{\mathfrak{p}} = v_i^{\mathfrak{q}}$, $N_i^{\mathfrak{p}}(-) = N_i^{\mathfrak{q}}(-)$, for $1 \le i \le r_0$. We claim that

$$v_i^{\mathfrak{q}}(\phi_{r+1}) = v_i^{\mathfrak{q}}(\phi_r^{e_r f_r}), \quad \text{for all } 1 \le i \le r_0 + 1. \tag{4.1}$$

Let us show this by induction; clearly, $v_1^{\mathfrak{q}}(\phi_{r+1}) = 0 = v_1^{\mathfrak{q}}(\phi_r^{e_r f_r})$, because both polynomials are monic. Suppose that (4.1) holds for some $1 \le i \le r_0$, and let us show that it holds for $i + 1$. By the definition of $v_{i+1}^{\mathfrak{q}}$, we need only to show that

$$N_i^{\mathfrak{q}}(\phi_{r+1}) = N_i^{\mathfrak{q}}(\phi_r^{e_r f_r}).$$

Since $i \le r_0$, it is sufficient to check that $N_i^{\mathfrak{p}}(\phi_{r+1}) = N_i^{\mathfrak{p}}(\phi_r^{e_r f_r})$. Now, these polygons are both one-sided of slope λ_i, and have the same length because the two polynomials, ϕ_{r+1}, $\phi_r^{e_r f_r}$, have the same degree. Finally, the two polygons have the same end point by the equality of (4.1) for our i, and by [HN, Lem.2.17]. This ends the proof of (4.1).

We claim now that

$$\omega_{r_0+1}^{\mathfrak{q}}(\phi_{r+1}) = \omega_{r_0+1}^{\mathfrak{q}}(\phi_r) = 0. \tag{4.2}$$

In fact, the polygon $N_{r_0}^{\mathfrak{q}}(\phi_r) = N_{r_0}^{\mathfrak{p}}(\phi_r)$ is one-sided of slope λ_{r_0}. If $\lambda_{r_0}^{\mathfrak{q}} \ne \lambda_{r_0}$, then $R_{r_0}^{\mathfrak{q}}(\phi_r)$ is a constant and $\omega_{r_0+1}^{\mathfrak{q}}(\phi_r) = 0$. If $\lambda_{r_0}^{\mathfrak{q}} = \lambda_{r_0}$, but $\psi_{r_0}^{\mathfrak{q}} \ne \psi_{r_0}$, then $R_{r_0}^{\mathfrak{q}}(\phi_r) = R_{r_0}^{\mathfrak{p}}(\phi_r)$ is a power of ψ_{r_0} up to a multiplicative constant, and $\psi_{r_0}^{\mathfrak{q}} \nmid R_{r_0}^{\mathfrak{q}}(\phi_r)$, so that $\omega_{r_0+1}^{\mathfrak{q}}(\phi_r) = 0$ too. The same argument works for ϕ_{r+1}.

Finally, (4.1) and (4.2) show that $v(\phi_{r+1}(\theta^{\mathfrak{q}})) = v(\phi_r(\theta^{\mathfrak{q}})^{e_r f_r})$. Therefore, $v_{\mathfrak{q}}(\phi_{r+1}(\theta)) = v_{\mathfrak{q}}(\phi_r(\theta)^{e_r f_r})$, and $v_{\mathfrak{q}}(\beta) = 0$.

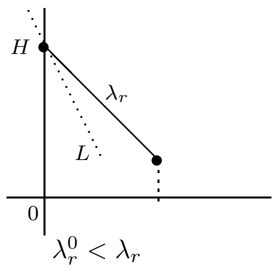
Assume now $r_0 = r$. The Main loop applied to $(\mathbf{t}_{\mathfrak{q}})_{r-1} = \mathbf{t}_{r-1}$ and the representative $\phi_r = \phi_r^{\mathfrak{p}}$ produced a complete branch $\mathbf{t}$, and (at least) another branch of order r

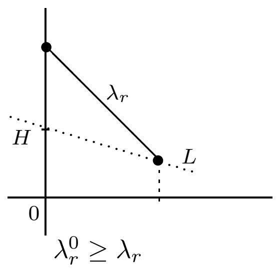
$$\mathbf{t}_{\mathfrak{q}}^0 = (\phi_1(x); \cdots; \lambda_{r-1}, \phi_r(x); \lambda_r^0, \psi_r^0(y)),$$

whose further development produced the type $\mathbf{t}_{\mathfrak{q}}$. Let us denote by v^0_{r+1}, ω^0_{r+1}, e^0_r the data attached to this type. Since $\theta^{\mathfrak{q}}$ is the root of a p-adic polynomial of type $\mathbf{t}^0_{\mathfrak{q}}$, the Theorem of the polygon shows that,

$$v(\phi_r(\theta^{\mathfrak{q}})) = (v_r(\phi_r) + |\lambda^0_r|)/e_1\cdots e_{r-1}. \tag{4.3}$$

Let us compute now $v(\phi_{r+1}(\theta^{\mathfrak{q}}))$. The Newton polygon $N^0_r(\phi_{r+1}) = N^{\mathfrak{p}}_r(\phi_{r+1})$ has length e_rf_r, slope λ_r, and the ordinate of the last point is $v_r(\phi_{r+1}) = v_r(\phi_r^{e_rf_r})$. Consider a line of slope λ^0_r far beyond the polygon, and move it upwards till it touches it; let L be this line of slope λ^0_r that first touches the polygon, and let H be the ordinate of L at the origin. We distinguish two cases, according to $\lambda^0_r < \lambda_r$ or $\lambda^0_r \ge \lambda_r$. In the first case, $\lambda^0_r = \lambda^{\mathfrak{q}}_{\mathfrak{p}}$ is the dominating slope of $\mathfrak{q}$ over $\mathfrak{p}$.





Arguing as in the case $r_0 < r$, we get $\omega^0_{r+1}(\phi_{r+1}) = 0$ in both cases, and

$$v(\phi_{r+1}(\theta^{\mathfrak{q}})) = v^0_{r+1}(\phi_{r+1})/e_1\cdots e_{r-1}e^0_r = H/e_1\cdots e_{r-1}, \tag{4.4}$$

the last equality by the definition of v^0_{r+1}. The figures show that $H = v_r(\phi_r^{e_rf_r}) + H'$, with

$$H' = \begin{cases} e_rf_r|\lambda_r|, & \text{if } \lambda^0_r < \lambda_r, \\ e_rf_r|\lambda^0_r|, & \text{if } \lambda^0_r \ge \lambda_r. \end{cases}$$

Now, (4.3) and (4.4), show that

$$\frac{v_{\mathfrak{q}}(\beta)}{e^{\mathfrak{q}}_1\cdots e^{\mathfrak{q}}_s} = \frac{v_r(\phi_r^{e_rf_r}) + H'}{e_1\cdots e_{r-1}} - \frac{v_r(\phi_r^{e_rf_r}) + e_rf_r|\lambda^0_r|}{e_1\cdots e_{r-1}}.$$

Therefore, $v_{\mathfrak{q}}(\beta) = 0$ if $\lambda^0_r \ge \lambda_r$, and otherwise,

$$v_{\mathfrak{q}}(\beta) = \frac{e^{\mathfrak{q}}_1\cdots e^{\mathfrak{q}}_s}{e_1\cdots e_{r-1}} e_rf_r(|\lambda_r| - |\lambda^{\mathfrak{q}}_{\mathfrak{p}}|).$$

□

For the maximal types with respect to the ordering ">", we take $\tilde{\alpha}_{\mathfrak{p}} := \beta_{\mathfrak{p}}$. For the rest of the types we compute recurrently:

$$\tilde{\alpha}_{\mathfrak{p}} := \beta_{\mathfrak{p}} \prod_{\mathbf{t}_{\mathfrak{q}} > \mathbf{t}_{\mathfrak{p}}} \tilde{\alpha}_{\mathfrak{q}}^{-v_{\mathfrak{q}}(\beta_{\mathfrak{p}})}.$$

These elements are not far from generating the $\mathfrak{p}_i$, since:

$$v_{\mathfrak{q}}(\tilde{\alpha}_{\mathfrak{p}}) = \begin{cases} 1 & \text{if } \mathfrak{q} = \mathfrak{p}, \\ 0 & \text{otherwise.} \end{cases}$$

Unfortunately, they could be non-integral at primes of $\mathbb{Z}_K$ not dividing $p\mathbb{Z}_K$. This can be easily arranged; we write each $\tilde{\alpha}_{\mathfrak{p}}$ in the form $\tilde{\alpha}_{\mathfrak{p}} = G(\theta)/b$, with $G(x) \in \mathbb{Z}[x]$ and $b \in \mathbb{Z}$ coprime with the content of $G(x)$; and we conveniently modify $\tilde{\alpha}_{\mathfrak{p}}$ into:

$$\alpha_{\mathfrak{p}} := G(\theta)/p^{v(b)}.$$

5. Computational issues

Recall that Montes' algorithm is the optimization of the Basic algorithm of section 2.2 that results from the application of the refinement process of section 3. In this section we give a more detailed description of this algorithm and we discuss some computational aspects. We also include, as an extension of the algorithm, the computation of generators for the prime ideals as indicated in the last section.

5.1. **Outline of the algorithm.**

The primary goal of Montes' algorithm is the computation of $\operatorname{ind}(f)$ and the construction of a set $\mathbf{T}$ of f-complete types, that faithfully represents $f(x)$.

By the recursive nature of its construction, many of the types generated by the algorithm will share many of its levels, so that most of the computations necessary to enlarge them will be the same. Hence it is very convenient to organize their computation in such a way that we can take profit of as much previous computations as possible. The simplest way to organize the computation of types is to store all the types being built by the algorithm in a list, which we call `STACK`. Once a type is completed, it is moved from the list `STACK` to a second list called `COMPLETETYPES`, which when the algorithm ends contains the final list of complete types.

The variable `STACK`, as its name suggests, is a LIFO stack, which in practice determines the flow of the algorithm: the main loop of the algorithm extracts the last type from the `STACK` and works it out to decide whether it is complete (in which case it is moved to `COMPLETETYPES`) or it originates a number of enlarged types, that will be added to the top of the `STACK`. The program finishes when the `STACK` is empty.

We could think of the types being built along the algorithm as the branches of a tree. The root of the tree is a node corresponding to the input polynomial $f(x)$. Every division of the branch into new subbranches is generated in every order by multiple sides of a Newton polygon of $f(x)$ and by multiple irreducible factors of the residual polynomial of each side. The algorithm builds this tree of types from the topmost branch to the lowest one in every order. This strategy confers a certain ordering to the list `COMPLETETYPES`, which is useful later on for the computation of generators of the ideals.

The computation of the p-index is performed along the construction of types: every time we analyze a Newton polygon with respect to a type $\mathbf{t}$, we add the number $\operatorname{ind}_{\mathbf{t}}^{h_r}(f)$, given in (3.2), to the variable `TOTALINDEX`, whose final output is the value of $\operatorname{ind}(f)$.

Once the algorithm has emptied the `STACK`, the algorithm is almost finished: it remains only to gather the information of every complete type to list the ramification indices and residual degrees of the prime ideals dividing $p\,\mathbb{Z}_K$.

We now give a detailed outline of Montes' algorithm, using standard pseudo-code.

MONTES' ALGORITHM

INPUT:

- A monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$.
- A prime number $p \in \mathbb{Z}$.

OUTPUT:

- The p-valuation of the index of $f(x)$ in $\mathbb{Z}_K$.
- A list $\{(e_1, f_1), \dots, (e_g, f_g)\}$ of pairs of integers describing the factorization of $p\,\mathbb{Z}_K$:

$$p\,\mathbb{Z}_K = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_g^{e_g}, \qquad f(\mathfrak{p}_i/p) = f_i.$$

- A list of integral elements $\alpha_1, \dots, \alpha_g \in \mathbb{Z}_K$ such that $\mathfrak{p}_i = p\,\mathbb{Z}_K + \alpha_i \mathbb{Z}_K$

INITIALIZATION STEPS

1 Factor $f(y) = \psi_1(y)^{a_1} \cdots \psi_s(y)^{a_s}$ modulo p, with $\psi_i(y) \in \mathbb{F}_p[y]$ pairwise different monic irreducible polynomials.

```
**2** Take monic polynomials $\phi_1(x),\dots,\phi_g(x)\in\mathbb{Z}[x]$ such that $\phi_i(y) \pmod p = \psi_i(y)$. Compute the polynomial $M(x)=(f(x)-\phi_1(x)^{a_1}\cdots\phi_s(x)^{a_s})/p$.
**3** Initialize empty lists STACK and COMPLETETYPES, and set TOTALINDEX$\leftarrow 0$.
**4** FOR every polynomial $\phi_i(x)$ do
    (Dedekind's criterion) If $a_i=1$ or $\psi_i(y)\nmid M(y) \pmod p$, output the ideal $\mathfrak{p}=(p,\,\phi_i(\theta))$, with ramification index $e(\mathfrak{p}/p)=a_i$ and residual degree $f(\mathfrak{p}/p)=\deg\phi_i$. Otherwise, add to STACK the extension $\tilde{\mathbf{t}}=(\phi_i(x))$ of the type of order zero determined by $\psi_i(y)$, and set $H_1=0$, as data of level one.

MAIN LOOP
WHILE the STACK is non-empty do:
**5** Extract the last type $\mathbf{t}^0$ from STACK. Let $r-1$ be its order.
**6** Compute the Newton polygon $N_r^{H_r}(f)$, formed by the sides of slope smaller than $-H_r$.
**7** Compute $\operatorname{ind}_{\mathbf{t}^0}^{H_r}(f)$ using the formula (3.2) and add this number to TOTALINDEX.
  FOR every side $S$ of $N_r^{H_r}(f)$ do
    **8** Set $\lambda_r^{\mathbf{t}^0}\leftarrow$ slope of $S$.
    **9** Compute the $r$-th order residual polynomial $R_r(f)(y)\in\mathbb{F}_r[y]$.
    **10** FOR every irreducible factor $\psi(y)$ of $R_r(f)(y)$ do
        **11** Make a copy $\mathbf{t}$ of the type $\mathbf{t}^0$, and set $\psi_r^{\mathbf{t}}(y)\leftarrow\psi(y)$.
        **12** Compute a representative $\phi_{r+1}(x)\in\mathbb{Z}[x]$ of $\mathbf{t}$.
        **13** If $\operatorname{ord}_\psi(R_r(f))=1$ (the type is complete), add $(\tilde{\mathbf{t}};\lambda_r^{\mathbf{t}},\psi^{\mathbf{t}}(y))$ to COMPLETETYPES, and continue to the next factor of $R_r(f)(y)$.
        **14** If $\deg\psi=1$ and $\lambda_r^{\mathbf{t}}\in\mathbb{Z}$ (the type must be refined), set $\phi_r^{\mathbf{t}}(x)\leftarrow\phi_{r+1}(x)$, $H_r^{\mathbf{t}}\leftarrow\lambda_r^{\mathbf{t}}$, add $\mathbf{t}$ to the top of the STACK and continue to the next factor of $R_r(f)(y)$.
        **15** (Build a higher order type) Add $(\tilde{\mathbf{t}};\lambda_r^{\mathbf{t}},\psi^{\mathbf{t}}(y))$ to the top of the STACK.
END OF MAIN LOOP

OUTPUT
**16** Print the $p$-valuation of the index of $f$ in $\mathbb{Z}_K$, given by the value of TOTALINDEX.
**17** For every type $\mathbf{t}$ in the list COMPLETETYPES, output the ramification index and residual degree of the corresponding ideal $\mathfrak{p}$, given by:
        $e(\mathfrak{p}/p)=e_1^{\mathbf{t}}\cdots e_r^{\mathbf{t}},\quad f(\mathfrak{p}/p)=m_1^{\mathbf{t}}f_1^{\mathbf{t}}\cdots f_r^{\mathbf{t}},$
    where $m_1^{\mathbf{t}}=\deg\phi_1^{\mathbf{t}}$, and $r$ is the order of $\mathbf{t}$

EXTENSION: COMPUTATION OF GENERATORS
All notations are taken from section 4
**18** FOR every type $\mathbf{t}_\mathfrak{p}$ in COMPLETETYPES compute the element $\beta_\mathfrak{p}=\tilde{\phi}_{r+1}(\theta)/\phi_r(\theta)^{e_rf_r}$.
**19** FOR every type $\mathbf{t}_\mathfrak{p}$ in COMPLETETYPES compute the element $\tilde{\alpha}_\mathfrak{p}:=\beta_\mathfrak{p}\prod_{\mathbf{t}_\mathfrak{q}>\mathbf{t}}\tilde{\alpha}_\mathfrak{q}^{-v_\mathfrak{q}(\beta_\mathfrak{p})}$, where $v_\mathfrak{q}(\beta_\mathfrak{p})$ is given in Proposition 4.2.
**20** FOR every type $\mathbf{t}_\mathfrak{p}$ in COMPLETETYPES compute the element $\alpha_\mathfrak{p}=G(x)/p^{v(b)}$.
```

To compute $\tilde{\alpha}_\mathfrak{p}$ in step **19**, it is necessary to slightly modify the algorithm in order to store all dominating slopes $\lambda_\mathfrak{p}^\mathfrak{q}$.

5.2. **Some examples.**

Example 1. Let us consider the irreducible polynomial

$$\begin{aligned} f(x) :=& x^{12} - 588x^{10} + 476x^9 + 130095x^8 - 172872x^7 - 12522636x^6 + 24745392x^5 \\ &+ 486721116x^4 - 1583408736x^3 - 641009376x^2 + 10978063488x + 59914669248, \end{aligned}$$

whose discriminant is

$$\operatorname{disc}(F) = 2^{84} \cdot 3^{64} \cdot 7^{52} \cdot 79^4 \cdot 14159^2 \cdot 644173^2 \cdot 3352073^2 .$$

We apply the algorithm to find the decomposition of the prime $p = 2$ in the ring of integers $\mathbb{Z}_K$ of the number field $K = \mathbb{Q}(\theta)$ generated by any root of the polynomial $f(x)$. Since

$$f(y) \equiv (y+1)^4\, y^8 \pmod 2,$$

we find two types $\mathbf{t}_1, \mathbf{t}_2$ of order zero, extended respectively to $\phi_1^1(x) = x + 1$, $\phi_1^2(x) = x$. The Newton polygon $N_1^{\mathbf{t}_1}(f)$ has two sides, with slopes $-3/2$ and $-1/2$ respectively, which single out two prime ideals $\mathfrak{p}_1, \mathfrak{p}_2$, with $e(\mathfrak{p}_1/2) = e(\mathfrak{p}_2/2) = 2$ and $f(\mathfrak{p}_1/2) = f(\mathfrak{p}_2/2) = 1$. The type $\mathbf{t}_{\mathfrak{p}_1}$ dominates $\mathbf{t}_{\mathfrak{p}_2}$ with dominating slope $\lambda_{\mathfrak{p}_2}^{\mathfrak{p}_1} = -3/2$.

The Newton polygon $N_1^{\mathbf{t}_2}(f)$ has again two sides, with slopes -1 and $-1/2$, and residual polynomials $R_{1,1}(f) = (y+1)^4$, $R_{1,2}(f) = (y+1)^2$, respectively. Hence, the type $\mathbf{t}_2$ yields two types $\mathbf{t}_{2,1}, \mathbf{t}_{2,2}$ of order one. The first one must be refined and the second one must be enlarged to an order 2 type. To refine $\mathbf{t}_{2,1}$, we take the new polynomial $\phi_1^{\mathbf{t}_{2,1}}(x) = x + 2$. The corresponding Newton polygon has only one side with slope smaller than -1; the slope is $-3/2$ and the residual polynomial $(y+1)^2$, so that this type must be enlarged too, to an order 2 type. After computing their respective representatives, we have now two extended types of order one, ready to be enlarged to order 2:

$$\begin{aligned} \mathbf{t}_{2,1} &= (x+2; -3/2, (x+2)^2 + 8), \qquad & H_1 = -1, \\ \mathbf{t}_{2,2} &= (x; -1/2, x^2 + 2), & H_1 = 0. \end{aligned}$$

The Newton polygon $N_2^{\mathbf{t}_{2,1}}(f)$ has a unique side with slope -4 and residual polynomial $(y+1)^2$, so that this type can be refined. We take, for instance, $\phi_2^{\mathbf{t}_{2,1}}(x) = (x+2)^2 + 40$:

$$\mathbf{t}_{2,1} = (x+2; -3/2, (x+2)^2 + 40), \quad H_1 = -1, \quad H_2 = -4.$$

The new polygon $N_2^{\mathbf{t}_{2,1}}(f)$ has two sides with slopes -9 and -5, originating two new prime ideals $\mathfrak{p}_3, \mathfrak{p}_4$ dividing $2\mathbb{Z}_K$, with $e(\mathfrak{p}_3/2) = e(\mathfrak{p}_4/2) = 2$, $f(\mathfrak{p}_3/2) = e(\mathfrak{p}_4/2) = 1$. The type $\mathbf{t}_{\mathfrak{p}_3}$ dominates $\mathbf{t}_{\mathfrak{p}_4}$ with dominating slope $\lambda_{\mathfrak{p}_4}^{\mathfrak{p}_3} = -9$.

The Newton polygon $N_2^{\mathbf{t}_{2,2}}(f)$ has a unique side with slope -4 and residual polynomial $(y+1)^2$; thus, we must refine. Take $\phi_2^{\mathbf{t}_{2,2}}(x) = x^2 + 10$, $H_2 = -4$. The next Newton polygon has again a unique side with slope -5 and residual polynomial $(y+1)^2$. We refine again, taking $\phi_2^{\mathbf{t}_{2,2}}(x) = x^2 + 10 + 8x$. We get:

$$\mathbf{t}_{2,2} = (x; -1/2, x^2 + 8x + 10), \quad H_1 = 0,\ H_2 = -5.$$

Now, $N_2^{\mathbf{t}_{2,2}}(f)$ has two sides with slopes -8 and -7, both with residual polynomial $y + 1$, thus giving two prime ideals $\mathfrak{p}_5, \mathfrak{p}_6$, with $e(\mathfrak{p}_5/2) = e(\mathfrak{p}_6/2) = 2$, $f(\mathfrak{p}_5/2) = e(\mathfrak{p}_6/2) = 1$. The type $\mathbf{t}_{\mathfrak{p}_5}$ dominates $\mathbf{t}_{\mathfrak{p}_6}$ with dominating slope $\lambda_{\mathfrak{p}_6}^{\mathfrak{p}_5} = -8$.

Summing up, $2\mathbb{Z}_K = (\mathfrak{p}_1 \cdots \mathfrak{p}_6)^2$. The 2-index of $f(x)$ is $\operatorname{ind}_2(f) = 33$, and $v(\operatorname{disc}(K)) = 18$.

Generators for the ideals $\mathfrak{p}_i$ can be determined by using the formulas of section 4. We find:

$$\begin{aligned} \tilde{\alpha}_1 &= \frac{(\theta+1)^2 + 4(\theta+1) + 8}{(\theta+1)^2}, & \tilde{\alpha}_2 &= \frac{(\theta+1)^2 + 2(\theta+1) + 2}{(\theta+1)^2}\,\tilde{\alpha}_1^4, \\ \tilde{\alpha}_3 &= \frac{(\theta+2)^2 + 64(\theta+2) + 40}{(\theta+2)^2 + 40}, & \tilde{\alpha}_4 &= \frac{(\theta+2)^2 + 16(\theta+2) + 40}{(\theta+2)^2 + 40}\,\tilde{\alpha}_3^4, \\ \tilde{\alpha}_5 &= \frac{\theta^2 + 40\,\theta + 42}{\theta^2 + 8\,\theta + 10}, & \tilde{\alpha}_6 &= \frac{\theta^2 + 24\,\theta + 10}{\theta^2 + 8\,\theta + 10}\,\tilde{\alpha}_5, \end{aligned}$$

$$\begin{aligned}
\alpha_1 &= (2\theta^{11}+\theta^{10}+2\theta^9-\theta^8+2)/2;\\
\alpha_2 &= (\theta^{11}+\theta^{10}+\theta^9-3\theta^8+4\theta^5+4\theta^4+4)/4;\\
\alpha_3 &= (131\theta^{11}-474\theta^{10}+448\theta^9+52\theta^8+309\theta^7-366\theta^6-216\theta^5+256\theta^4-364\theta^3\\
&\qquad +136\theta^2-496\theta+32)/512;\\
\alpha_4 &= (-27\theta^{11}-2\theta^{10}-80\theta^9+12\theta^8+19\theta^7+10\theta^6+72\theta^5-64\theta^4+76\theta^3\\
&\qquad -104\theta^2+48\theta+32)/128;\\
\alpha_5 &= (-45\theta^{11}-10\theta^{10}-18\theta^9+64\theta^8-7\theta^7+50\theta^6-30\theta^5-60\theta^4+32\theta+64)/64;\\
\alpha_6 &= (33\theta^{11}+8\theta^{10}+42\theta^9-4\theta^8-53\theta^7-8\theta^6-58\theta^5+16\theta^4+32\theta^3+64\theta^2-32\theta)/64.
\end{aligned}$$

Example 2. Take $p=2$ and consider the irreducible polynomial

$$f(x) := (x^3+x+5)^{50}+2^{89}(x^3+x+5)^{25}+2^{178}\,.$$

The algorithm takes initially $\phi_1(x)=x^3+x+1$, and finds a unique side with slope -2 and residual polynomial $(y+1)^{50}$. A refinement leads to $\phi_1(x)=x^3+x+5$, and a Newton polygon with one side, with slope $-89/25$ and irreducible residual polynomial y^2+y+1. Hence, in the number field K defined by any root of $f(x)$, we have $2\mathbb{Z}_K=\mathfrak{p}^{25}$, with $f(\mathfrak{p}/2)=6$. The 2-index of the polynomial is 13011. While this computation is almost instantaneous, the determination with Pari of a 2-integral basis of K takes about 190 seconds, and needs an amount of 244 Mb of memory.

5.3. **Some remarks on the complexity.** We have not developed a detailed analysis of the complexity of the algorithm, but the experimental results of the next section indicate that its running time is excellent. We now provide some arguments to explain this good behaviour.

We saw in section 2.3 that the number of iteration of the Main loop is bounded by $\operatorname{ind}(f)$ and that each iteration covers $\operatorname{ind}_{\mathbf{t}}(f)$ steps from the total value of $\operatorname{ind}(f)$. One is tempted to conclude that the running time of the algorithm is linear on the discriminant of $f(x)$, but this is not so evident, since the treatment of higher order types is much expensive than the treatment of low order types. However, this is balanced by two facts: on one hand, $\operatorname{ind}_{\mathbf{t}}(f)$ is generally much bigger than one in each iteration; on the other hand, the higher the order, the smaller is $\omega_{r+1}(f)$, and this invariant tells the number of coefficients of the ϕ_r-adic development of $f(x)$ that must be computed, the length of the Newton polygon to be analyzed, and it is an upper bound for the degrees of the residual polynomials.

At least, it seems that for polynomials whose types have bounded order, the running time will be at most linear on the discriminant. In practice, the average running time is much smaller.

On the other hand, the degree of the polynomials $\phi_k^{\mathbf{t}}(x)$ appearing in an f-complete type $\mathbf{t}$ is a divisor of the product $e(\mathfrak{p}_{\mathbf{t}}/p)f(\mathfrak{p}_{\mathbf{t}}/p)$. Every time the type is enlarged, the degree of the last $\phi_k(x)$ is multiplied by the corresponding product $e_k^{\mathbf{t}}f_k^{\mathbf{t}}$. Hence, for a polynomial $f(x)$ to have attached a type of a very high order, its degree must be really huge. This explains why the algorithm works well for polynomials of high degree: the maximum of the orders of the types of a polynomial grows slowly in comparison with the degree.

The low memory requirement of the algorithm is another of its strong advantages: it is only necessary to store the polynomials $\phi_k(x)$ and $\psi_k(x)$ of the types being calculated (and some Taylor expansions to gain efficiency). This makes possible the treatment of polynomials of very high degree with scarce computational resources.

The complexity of the computation of the generators is dominated by the inversion of $\phi_r(\theta)$ in K, which is a hard task if the degree of $\phi_r(x)$ is large.

5.4. **Implementation of the algorithm.** The first implementation of Montes' algorithm was programmed by J. Guàrdia in 1997, as a part of his Ph.D. It was written for Mathematica 3.0, and it included a specific package to work with finite fields, since that version of Mathematica did not carry such a package. It is still available on request to the author. Ten years later, we started a collaboration to make a full upgrade of the algorithm, with many optimizations both theoretical and computational, including a completely new implementation in Magma.

The computation of generators for the prime ideals becomes a heavy time-consuming task if the p-adic factors of $f(x)$ have large degrees. For this reason, our implementation skips this calculation by defect. If the user wants to compute the generators, a Boolean variable `GENERATORS` has to be given the value "true".

As memory requirement is not a constraint for the implementation, the program stores some intermediate results (mainly ϕ-adic expansions) to gain speed. The main data type used by the program is a specifically designed record which contain all the relevant data of a type in a given order. To avoid massive replication of the types being computed, must of the routines access them by memory address.

The program also includes a number of routines to construct types and polynomials with a prescribed set of types. The program and its documentation, which includes all the examples presented in this paper, can be downloaded from the web page

`www-ma4.upc.edu/~guardia/MontesAlgorithm.html`.

Any comment on the program will be welcome.

6. Some heuristics on the complexity

We dedicate this section to illustrate the performance of (our implementation of) Montes' algorithm with several polynomials chosen to force its capabilities at maximum in three directions: polynomials with a unique associate type of large order, polynomials which require a lot of refinements, and polynomials with many different types. We have also included some test polynomials found in the literature.

All the tests have been done in a personal computer, with an Intel Core Duo processor, running at 2.2 Mhz, with 3Gb of RAM memory. The reader willing to check these results on his/her own can obtain the Magma code to generate these polynomials from the web page of the program.

Example 1: Take $p = 2$. Consider the irreducible polynomials

$$\begin{aligned}
\phi_1 &= x^2 + 2^2x + 2^4;\\
\phi_2 &= \phi_1^2 + 2^4x\phi_1 + 2^{12};\\
\phi_3 &= \phi_2^4 + 2^{23}(x+2^2)\phi_2^2 + 2^{42}x\phi_1;\\
\phi_4 &= \phi_3^2 + 2^{12}x\phi_2^3\phi_3 + 2^{72}\phi_1\phi_2^2 + 2^{101}x;\\
\phi_5 &= \phi_4^3 + 2^{34}\phi_1\phi_2\phi_3\phi_4^2 + 2^{215}((x(\phi_1+2^6)(\phi_2^3+2^{25}\phi_2) + 2^{27}\phi_2)\phi_3 + 2^{64}(x\phi_1\phi_2^2+2^{33}));\\
\phi_6 &= \phi_5^6 + 2^{883}x\phi_3\phi_5^3 + 2^{1736}((x+4)\phi_1 + 2^8)\phi_2^2\phi_4;\\
\phi_7 &= \phi_6^2 + 2^{2351}((\phi_1\phi_2^3 + 2^{23}x(\phi_1+2^6)\phi_2)\phi_4 + 2^{102}(x\phi_1\phi_2^3 + 2^{25}((x+2^2)\phi_1 + 2^6x)\phi_2))\phi_5^4\\
&\quad +2^{3234}(((x\phi_1\phi_2^3 + 2^{25}(x+2^2)(\phi_1+2^6)\phi_2)\phi_3 + 2^{70}(x\phi_2^2+2^{27}))\phi_4\\
&\quad +2^{168}((x+2^2)\phi_1 + 2^6x)\phi_2^2)\phi_5;\\
\phi_8 &= \phi_7^6 + 2^{7515}((((((x+4)\phi_1 + 2^6x)\phi_2^2 + 2^{31}x)\phi_3 + 2^{39}x\phi_1\phi_2^3 + 2^{70}x\phi_2)\phi_4^2\\
&\quad +2^{104}(((x\phi_1+2^8)\phi_2^2 + 2^{25}(x\phi_1 + 2^6(x+2^4)))\phi_3 + 2^{38}((x+4)(\phi_1+2^6)\phi_2^3 + 2^{27}\phi_1\phi_2))\phi_4\\
&\quad +2^{208}(((x\phi_1+2^8)\phi_2^2 + 2^{25}x\phi_1 + 2^{31}(x+4))\phi_3 + 2^{41}\phi_1\phi_2^3 + 2^{64}(x\phi_1+2^8)\phi_2))\phi_5^5\\
&\quad +2^{924}(((((x+4)\phi_1 + 2^6x)\phi_2^3 + 2^{25}((x+4)\phi_1+2^8)\phi_2)\phi_3\\
&\quad +2^{134}((\phi_1 + 2^4(x+4))\phi_2^2 + 2^{25}(\phi_1+2^4x)))\phi_4^2\\
&\quad +2^{104}(((x+4)\phi_1\phi_2^3 + 2^{25}x(\phi_1+2^6)\phi_2)\phi_3 + 2^{64}((x+4)(\phi_1+2^6)\phi_2^2 + 2^{27}\phi_1))\phi_4\\
&\quad +2^{210}(((\phi_1+2^4x)\phi_2^3 + 2^{23}(x+4)(\phi_1+2^6)\phi_2)\phi_3 + 2^{64}(\phi_1\phi_2^2 + 2^{25}\phi_1)))\phi_5^2)\phi_6\phi_7^3\\
&\quad +2^{20618}(((x\phi_1\phi_2^3 + 2^{31}(x+2^6)\phi_2)\phi_3 + 2^{66}(\phi_1\phi_2^2 + 2^{25}(\phi_1+2^6)))\phi_4^2\\
&\quad +2^{104}(((x+4)\phi_1\phi_2^3 + 2^{25}x\phi_1\phi_2)\phi_3 + 2^{70}((x+4)\phi_2^2 + 2^{21}\phi_1))\phi_4\\
&\quad +2^{208}((((x+4)\phi_1 + 2^6x)\phi_2^3 + 2^{25}((x+4)\phi_1+2^8)\phi_2)\phi_3 + 2^{70}(x\phi_2^2 + 2^{19}(x\phi_1+2^8))))\phi_5^5
\end{aligned}$$

$$+2^{21567}(((x(\phi_1+2^6(x+4))\phi_2^2+2^{25}(x+4)\phi_1)\phi_3+2^{39}((x\phi_1+2^8)\phi_2^3+2^{25}(x+4)\phi_1\phi_2))\phi_4^2$$
$$+2^{104}((((x+4)\phi_1+2^6x)\phi_2^2+2^{33})\phi_3+2^{64}x\phi_1\phi_2)\phi_4$$
$$+2^{208}(x+4)(\phi_1+2^6)\phi_2^2\phi_3+2^{249}(\phi_1+2^4x)\phi_2^3)\phi_5^2.$$

For each j, the corresponding polynomial ϕ_j has a unique associate complete type of order j, so that in the corresponding number field K_j the ideal $2\mathbb{Z}_{K_j}$ is the power of a unique prime ideal $\mathfrak{p}_j$. The following table contains the degree and 2-index of the ϕ_k, the ramification index e_j and residual degree f_j of $\mathfrak{p}_j$ and the time $\mathtt{t}_1$ used by the program to compute them. The last column indicates the time $\mathtt{t}_2$ to run the extended version of the algorithm, which includes the computation of generators for the ideals $\mathfrak{p}_j$. All the times are expressed in seconds. The computation of the generators in the last two rows was stopped after 24 hours of running time.

ϕ_j	$\deg\phi_j$	$\mathrm{ind}(\phi_j)$	e_j	f_j	$\mathtt{t}_1$	$\mathtt{t}_2$
ϕ_1	2	2	1	2	0.00	0.00
ϕ_2	4	16	1	4	0.01	0.01
ϕ_3	16	360	2	8	0.01	0.01
ϕ_4	32	1544	2	16	0.01	0.016
ϕ_5	96	14616	2	48	0.08	0.6
ϕ_6	576	537120	6	96	0.70	406
ϕ_7	1152	2153376	12	96	4.0	
ϕ_8	6912	77673504	36	192	787	

Example 2: Let $f^k(x)=(x^2+x+1)^2-p^{2k+1}$, with $p\equiv 1 \pmod 7$ a prime number. When we apply Montes'algorithm to factor the ideal $p\,\mathbb{Z}_K$, we obtain two types of order zero with liftings $\phi_1(x)\in\mathbb{Z}[x]$ of degree one. For both of them the Newton polygon has only one side, with slope -1 and end points $(2,0)$ and $(0,2)$, and the residual polynomial is the square of a linear factor. After approximately $2k$ total refinements, both types become f^k-complete. The ideal $p\,\mathbb{Z}_K$ splits as the product of two prime ideals with ramification index 2 and residual degree 1, and the p-index of $f^k(x)$ is $2k$.

This is almost the illest-conditioned quartic polynomial for the algorithm, since the index of every type is increased a unit per refinement in general, and the total p-index of $f^k(x)$ is $2k$. Thus, the program has to make about $2k$ iterations of the main loop. Numerical experimentation shows that even in this worst case the running time of the algorithm is very low. In the following table we show the running time of the programm for different values of k and p. As before, $\mathtt{t}_1$ is the time in seconds to compute the index, residual degrees and ramification indices, and $\mathtt{t}_2$ is the time to compute also the generators for the prime ideals.

p	$\mathrm{ind}(f^k)$	$\mathtt{t}_1$	$\mathtt{t}_2$	p	$\mathrm{ind}(f^k)$	$\mathtt{t}_1$	$\mathtt{t}_2$
7	1000	0.57	0.62	43	10000	229	237
7	2000	1.95	2.1	103	10000	324	334
7	4000	8.7	9.2	1009	1000	2.1	2.6
7	8000	44.7	46.2	1009	2000	10.8	12.3
7	16000	245	250	1009	4000	58	62
7	20000	436	444	10^9+9	1000	10	12.7
13	1000	0.75	0.85	10^9+9	2000	57	66.5
13	2000	2.9	3.1	10^9+9	4000	313	341
13	10000	131	135	$10^{69}+9$	100	2.8	4.8
19	10000	158	162	$10^{69}+9$	200	8	13.5
31	10000	198	205	$10^{69}+9$	400	29.4	48
37	10000	214	221	$10^{69}+9$	1000	221	308

Example 3: Take $p = 13$. We now consider a polynomial with several different types. Let

$$\begin{aligned}
&\phi_1(x) = x^2 + 13^2 x + 13^4 \cdot 3;\\
&\phi_2(x) = \phi_1(x)^3 + ((13^{18} \cdot 2));\\
&\phi_3(x) = \phi_2(x)^{10} + 13^{89}(x + 13^2)\phi_2(x)^5 + 13^{176}\phi_1(x);\\
&\phi_4(x) = \phi_3(x)^2 + 13^{248}(12(x + 13^2)\phi_1(x) + 13^8)\phi_2(x)^6 + 13^{335} \cdot 12\phi_1(x)^2\phi_2(x);\\
&f_j(x) = \prod_{k=0}^{j} \phi_4(x + k) + 13^{5000}, \qquad j = 0, \ldots, 12;
\end{aligned}$$

In the number field K_j defined by $f_j(x)$, we have the factorization

$$13\mathbb{Z}_L = \mathfrak{p}_1^5 \cdots \mathfrak{p}_j^5, \qquad f(\mathfrak{p}_j/13) = 24.$$

Each prime ideal comes from a different order 4 type. The 13-index of $f_j(x)$ is $21576j$. The times to compute this index and the factorization of 13 in the fields K_j are shown in the table below.

j	$\deg f_j$	$\operatorname{ind}(f_j)$	$\mathtt{t}_1$
1	120	21576	0.08
2	240	43152	0.3
3	360	64728	0.9
4	480	86304	2.3
5	600	107880	4.4
6	720	129456	8.2
7	840	151032	13.3
8	960	172608	20.3
9	1080	194184	29.4
10	1200	215760	40.6
11	1320	237336	55.2
12	1440	258912	72
13	1560	280488	92

The computation of the generators for the number field defined by polynomial $f_1(x)$ took 20 seconds, for the number field defined by $f_2(x)$ lasted 6 1/2 hours. The computation for f_3 exhausted Magma's virtual memory due to a coefficient explosion in the computation of an extended gcd.

In this example one cannot expect a linear behaviour of the time versus the number of types, because the addition of more factors to the product defining the $f_j(x)$ implies a significant growing in the size of the coefficients of the polynomial, which has a certain impact in the running time of the algorithm.

Example 4: We applied the algorithm to the list of 32 polynomials $f_1, \ldots, f_{32}$ appearing in [FPR02, appendix D]. The total running time for altogether was less than 0.2 seconds. We then applied the algorithm to the polynomials $F_i = f_i^2 + p_i^{1000}$, where p_i is the prime specified in *loc.cit.* for every polynomial. In the table below we display the index of these polynomials and the running times of the algorithm. As before, $\mathtt{t}_1$ denotes the time in seconds to determine the index, the residual degrees and the ramification indices, and $\mathtt{t}_2$ is the time of the extended algorithm that includes the computation of the generators of the ideals.

f	p	ind(f)	$\mathtt{t}_1$	$\mathtt{t}_2$	f	p	ind(f)	$\mathtt{t}_1$	$\mathtt{t}_2$
F_1	2	2502150	2.45	3.1	F_{17}	2	1571054	2.9	7
F_2	2	1481141	2	2.7	F_{18}	7	7331055	77	93.9
F_3	3	2570992	6.6	8.2	F_{19}	71	187219	116.1	249.3
F_4	3	1177569	4.6	7.5	F_{20}	3	287752	15.4	23.2
F_5	2	2502505	1	1.5	F_{21}	5	10117231	73.8	81.2
F_6	2	2137558	1.6	2.4	F_{22}	3	5194476	18.4	23.6
F_7	2	2751159	3	4.7	F_{23}	3	2888852	15.7	23.6
F_8	5	1646099	13.81	20.5	F_{24}	2	2901708	6.2	9.5
F_9	2	1672713	2	3	F_{25}	47	2612660	253.5	636
F_{10}	1289	1500768	117	234	F_{26}	61	4257732	158	192
F_{11}	2	2629928	3	4.1	F_{27}	2	5925350	7.7	9.4
F_{12}	3	5895414	20	22.6	F_{28}	3	5720164	5	7.1
F_{13}	11	1810788	35	64.6	F_{29}	3	7826660	15.8	23
F_{14}	17	1618581	31.7	58	F_{30}	2	39363539	14.7	26.3
F_{15}	2	7744913	4.9	6.1	F_{31}	2	40933692	62.4	73
F_{16}	2	3808558	4.3	6.9	F_{32}	2	17097775	82.7	132

References

[BL94] Buchmann, J.A.; Lenstra, H.W., *Approximating ring of integers in number fields*, J. Théorie des Nombres de Bordeaux, 6, no. 2 (1994), pp. 221–260.

[Coh00] Cohen, H., *A course in Computational Number Theory*, Graduate Texts in Mathematics, vol. 138, Springer V., Berlin, 2000, fourth edition.

[FL94] Ford, D.; Letard, P., *Implementing the Round Four maximal order algorithm*, J. Théorie des Nombres de Bordeaux, 6, no. 1 (1994), pp. 39–80.

[FPR02] Ford, D.; Pauli, S.; Roblot, X., *A fast algorithm for polynomial factorization over* $\mathbb{Q}_p$, J. Théorie des Nombres de Bordeaux, 14, no. 1 (2002), pp. 151–169.

[HN] Guàrdia, J.; Montes, J.; Nart, E., *N*ewton polygons of higher order in algebraic number theory, arXiv:0807.2620[math.NT].

[Mon99] Montes, J., *Polígonos de Newton de orden superior y aplicaciones aritméticas*, Tesi Doctoral, Universitat de Barcelona 1999.

[PZ89] Pohst, M.; Zassenhaus, H., *Algorithmic Algebraic Number Theory*, Cambridge Univ. Press, Cambridge (1989).

Departament de Matemàtica Aplicada IV, Escola Politècnica Superior d'Enginyera de Vilanova i la Geltrú, Av. Víctor Balaguer s/n. E-08800 Vilanova i la Geltrú, Catalonia
E-mail address: guardia@ma4.upc.edu

Departament de Ciències Econòmiques i Socials, Facultat de Ciències Socials, Universitat Abat Oliba CEU, Bellesguard 30, E-08022 Barcelona, Catalonia, Spain
E-mail address: montes3@uao.es

Departament de Matemàtiques, Universitat Autònoma de Barcelona, Edifici C, E-08193 Bellaterra, Barcelona, Catalonia
E-mail address: nart@mat.uab.cat